From Strings to Semantics: A Graph-based Reranking Approach for Annotating Tables using Domain Ontologies*

Nan Liu[1,*], Mohamed-Anis Koubaa[1], Wolfgang Suess[1] and Veit Hagenmeyer[1]

[1]Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany

Abstract

As one of the most widely used data storage and exchange formats, tabular data can be challenging to be integrated, interpreted, and reused when they lacks accurate semantic annotations, particularly when data come from heterogeneous sources. However, the annotation process is often time-consuming and requires a deep understanding of the internal structure of the target ontology. Therefore, developing efficient and accurate semi-automatic or fully automatic annotation tools is very important. Most existing approaches often rely on textual similarity to match column headers to ontology terms, and fail to effectively leverage the rich relational semantics representation within the ontology. To address this issue, we propose a reranking approach that combines semantic similarity with ontology structure. Specifically, we first generate a set of candidate ontology terms based on semantic similarity. For each source table header and its candidate ontology terms, we construct subgraphs and train a lightweight Graph Neural Network (GNN) model on these graphs to learn structure-aware representations. These representations are then used to improve the ranking of candidate ontology terms. To validate our approach, we performe experiments on the OAEI dataset. The results demonstrate that our approach improves Hit@1 by 4% compared to a baseline model that only relies on lexical similarity. This result shows that learning on local subgraphs is a promising direction for ontology alignment and schema matching.

Keywords

Graph Neural Networks, Information Retrieval, Reranking, Semantic Annotation, Ontology Matching, Natural Language Processing

1. Introduction

Interoperability and knowledge integration between heterogeneous data sources have always been key challenges in the semantic web domain. A large amount of tabular data are often generated and stored in separate databases across different infrastructures. The semantics of such data are always ambiguous and non-standardized, which impedes the implementation of the FAIR principles [1]. However, annotating tabular data is not a simple task. It is time-consuming, error-prone, and requires a deep understanding of the target ontology. The task of mapping table headers to ontology terms can be treated as a data matching problem [2]. Previous research has proposed various approaches, such as [3, 4, 5]. Recently, Large Language Models (LLMs) and Pre-trained Language Models (PLMs) like Sentence-Bidirectional Encoder Representation Transformer (SBERT) [6] have been widely used for data matching tasks. These models can capture contextual meaning and have shown promising results [7, 8]. However, most of them rely only on lexical or contextual similarity and are therefore incapable of reasoning about complex relationships defined in OWL axioms, such as hierarchies, subclass relations, and property dependencies. This limitation becomes more significant in domain-specific tasks. In addition, some LLM-based methods [9, 10] have demonstrated strong performance in zero-shot annotation tasks, but their decision-making processes are difficult to explain due to their black-box nature [8].

Inspired by recent research in the application of Graph Neural Networks (GNNs) to knowledge graph completion and reranking tasks [11, 12, 13], we propose a lightweight reranking approach that integrates ontology structure into the matching process. We construct a subgraph for each source table header

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan

*Corresponding author.
nan.liu@kit.edu (N. Liu); mohamed.koubaa@kit.edu (M. Koubaa); wolfgang.suess@kit.edu (W. Suess); veit.hagenmeyer@kit.edu (V. Hagenmeyer)
0009-0005-8768-7072 (N. Liu); 0000-0001-8552-2008 (M. Koubaa); 0000-0003-2785-7736 (W. Suess); 0000-0002-3572-9083 (V. Hagenmeyer)

and its candidate ontology terms, and train a GNN model on these graphs. By passing and aggregating messages among nodes, the model evaluates both the semantic and structural similarity and generates the final ranking of candidate terms. The proposed approach significantly reduces computational cost and improves annotation accuracy. The main contributions of this paper are as follows:

- We propose an approach for dynamically constructing context graphs for semantic annotation and reranking tasks, which improves the matching accuracy and enhances computational efficiency;
- We perform our approach on several real-world datasets, the approach achieves significant performance gains over the baseline model and is able to generate higher quality semantic annotations.

The remainder of this paper is structured as follows: Section 2 shows related work on graph-based reranking techniques. Section 3 introduces the proposed methodology. Section 4 describes the experimental setup and results. Section 5 concludes the paper with future work.

2. Related Work

Previous work [12] shows that graph-learning methods can be effectively used for reranking and Retrieval Augmented Generation (RAG) tasks. In graph-based reranking approaches, candidate documents are modeled as nodes, and candidate-candidate edges are constructed from semantic similarity and external knowledge. Then, the message passing or aggregation will be used for structured reasoning within the candidate set to generate more reliable candidates. The training methods of graph-based reranking models can be categorized into three types [14], they are point-wise [15, 16], pair-wise [17, 18], and list-wise [19]. Motivated by these works, we adapt this idea to Column Type Annotation (CTA) tasks. We construct subgraphs for each table header, where nodes of the subgraph are the top-K candidate ontology terms, and edges are derived from semantic similarity between candidates and structural relations in the target ontology (such as subClassOf, part_Of, has_quality). A graph-based reranking model then scores the nodes on this subgraph to get the final ranking.

3. Methodology

In this section, we provide a brief description of our approach and its implementation details. As shown in Figure 1, the proposed approach can be divided into two stages. In the first stage, we use an SBERT to retrieve the top-K candidate ontology terms based on semantic similarity. In the second stage, we construct a local subgraph for each table header and its candidate ontology terms. These subgraphs are then used as input to a GNN, which learns structure-aware representations to rerank the candidate ontology terms. In the following, we first define the problem formally and then describe in detail how the subgraphs are constructed.

3.1. Problem Formulation

Our task can be defined as: Given a target ontology $\mathcal{O}$ that contains a set of terms $T_{\mathcal{O}} = \{t_1, t_2, \ldots, t_m\}$ and an input table header h, we first apply an SBERT bi-encoder $\phi(\cdot)$ to obtain embeddings $e_h = \phi(h)$ and $e(t_i) = \phi(t_i)$, compute cosine similarities $s(h, t_i) = \langle e_h, e(t_i) \rangle / (\|e_h\| \, \|e(t_i)\|)$, and return a top-$K$ candidate list $L_{\text{cand}} = \left[(t_{c_1}, s_{c_1}), \ldots, (t_{c_K}, s_{c_K})\right]$ sorted by similarity score s. Then for each table header h we construct a header-specific candidate subgraph $G_h = (N, E, W)$: the node set $N = \{h_i, t_{c_1}, \ldots, t_{c_K}\}$ contains the candidates for that header h, edges and weights (E, W) capture pairwise relatedness, for example, semantic similarity or ontology relations. Each node has features $x_i = \left[\, e(t_{c_i});\, s_{c_i} \,\right]$ and embeddings e_h. Then we define a reranking function f_{rerank} based on a pre-trained GNN model. This function takes the table header h and the candidate list $L_{cand} = \{t_{cand_1}, t_{cand_2}, ..., t_{cand_k}\}$ as input, and outputs final results as $L_{rerank} = \{t_{rerank_1}, t_{rerank_2}, ..., t_{rerank_k}\}$.

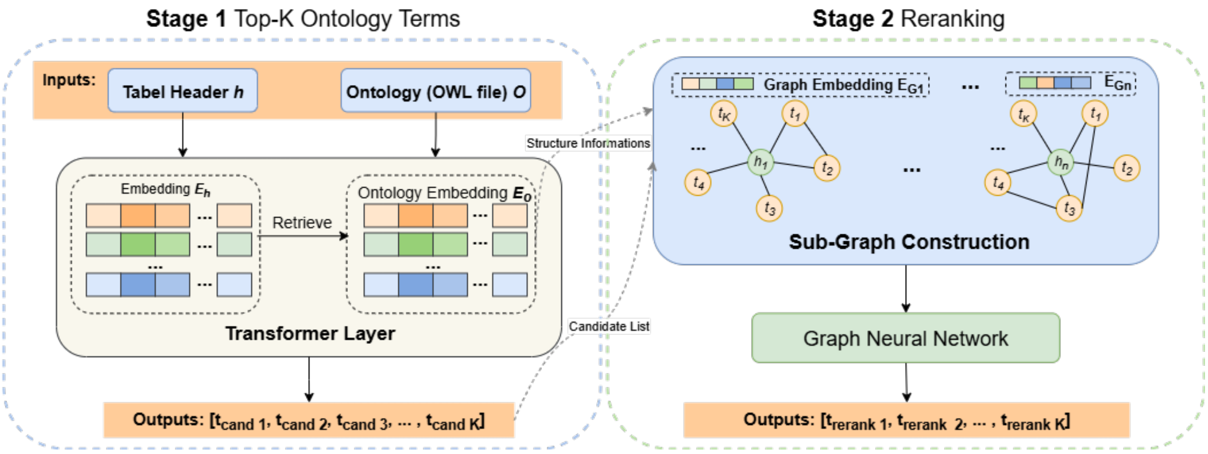

Figure 1: Proposed graph-based annotation approach with two stages: **Stage 1** generates top-K candidate ontology terms with SBERT. **Stage 2** uses GNN to rerank candidate ontology terms.

3.2. Graph Construction

To enable the reranking process, we construct a subgraph for each source header h_i and its candidate ontology terms $L_{cand} = \{t_{cand_1}, t_{cand_2}, ..., t_{cand_k}\}$, which are shown in Algorithm 1. We represent h_i and $L_{cand} = \{t_{cand_1}, t_{cand_2}, ..., t_{cand_k}\}$ as nodes in a graph. To connect the nodes, we add edges between the h_i and each candidate t_{cand}. The edge weights are the semantic similarity score calculated from the first-stage retrieval. In order to include structural information of the target ontology O, we search for the relations between candidate terms $L_{cand} = \{t_{cand_1}, t_{cand_2}, ..., t_{cand_k}\}$ in the target ontology O (such as `subClassOf`, `part_of`, `has_quality`). If the relation r exists and (t_i, r, t_j) is true, we add an edge between t_i and t_j. Each edge contains two features: a similarity score and the binary value of whether it is a structural edge of the ontology. In addition, we add self-loop edges to all nodes with a fixed weight of 1. These self-loops help preserve their own node features during message propagation.

Algorithm 1: Graph Construction

Input: Header text h, ontology O, candidate list L_{cand}
Output: Graph $G = (N, E)$ with node features and edge weights
`// Step 1: Graph Nodes`
$N \leftarrow \{h_i, t_1, \ldots, t_K\}$, where h_i is the source table header and $\{t_1, ..., t_k\}$ are the candidate ontology terms.
`// Step 2: Add Edges to Graph`
1. Add source-to-candidate edges (h_i, t_i) with edge feature $[sim_i, 0]$
2. Add candidate-to-candidate edges (t_i, t_j) with feature $[sim, \text{is\_ontology}]$
3. Add self-loops (h_i, h_i) with feature $[1, 0]$
return $G = (N, E)$ *with node features and edge weights*

3.3. Model Training

To learn structural representation for re-ranking candidate terms, we train a lightweight Graph Attention Model (GAT) based on GATv2 [20]. The GAT model consists of two GATv2 convolutional layers, followed by a linear classifier. In the training process, we use the RankNet loss [21]:

$$L_{ij} = \log\left(1 + e^{-\sigma(s_i - s_j)}\right) \tag{1}$$

For each graph, we sample all positive s_i and negative s_j candidate pairs and compute the average pairwise ranking loss. The goal is to rank the correct term as high as possible in the final reranking list L_{rerank}.

Table 1
Experimental results based on two OAEI datasets

Reranking Model	Hit@1	Hit@5	Hit@10	MRR
Baseline (SBERT only)	0.782	0.896	0.921	0.828
Rerank with MMR	0.785	0.899	0.921	0.834
Rerank with CE	0.808	0.905	0.921	0.846
Rerank with GCN	0.629	0.882	0.921	0.734
Rerank with GAT	**0.824**	**0.914**	0.921	**0.863**

4. Experiment and Results

4.1. Experiment Setup

We conduct experiments on the Bio-ML track [1] of the OAEI (Ontology Alignment Evaluation Initiative) benchmark in 2024, which focuses on ontology alignment tasks in the biomedical domain. The dataset[2] used in our experiments consists of three parts:

- **Source Header:** Each class label in the source ontology is treated as a source header to be annotated. We use the NCIT ontology as the source ontology in this experiment.
- **Target Ontology:** The complete target ontology is used for candidate retrieval. We select DOID as the target ontology.
- **Ground Truth Dataset:** The official reference alignment file with a unique correct match in the target ontology for each source header.

We use two standard ranking metrics to evaluate the performance of the model: Hit@K to evaluate top-K accuracy and Mean Reciprocal Rank (MRR) to evaluate the overall quality of the reranked results [22]. We evaluate all methods on the same candidate set generated by the first-stage SBERT bi-encoder. The systems compared are as follows: SBERT-only, using the first-stage similarity score as the final score; A non-graph reranker based on Maximal Marginal Relevance (MMR) that post-processes the SBERT list to balance relevance and diversity; A lightweight Cross-Encoder (CE) that concatenates the table header with candidate terms and inputs them into a single transformer and rescoring relevance to generate the final score; And two graph-based reranking models, Graph Convolutional Neural Network (GCN) and GAT that operate on the candidate subgraph.

4.2. Results and Analysis

Preliminary experimental results are shown in Table 1. The proposed GAT model achieves the best overall performance. It reaches a Hit@1 of 0.824, with an accuracy improvement of 4% over the SBERT-only baseline model (Hit@1 of 0.782), and GCN model (Hit@1 of 0.629). In addition, the GAT model also achieves the highest MRR score of 0.863. The results demonstrate the effectiveness of incorporating ontology structure into the reranking process and highlight the significant potential for enhancing schema matching tasks.

5. Conclusion and Future Works

In this paper, we propose a graph-based reranking approach, which improves the performance of semantic annotation tasks. By constructing a local subgraph for each table header and its candidate ontology terms, our method effectively integrates lexical semantic similarities with structural knowledge. Experiments on the OAEI Bio-ML track dataset show that our approach results in a Hit@1 of 0.824

[1]https://krr-oxford.github.io/OAEI-Bio-ML/
[2]https://zenodo.org/records/13119437

and a 4% improvement compared to the baseline model. These results provide a new perspective on performing efficient annotation solutions with reduced computational and cost demands.

For future work, we plan to enrich the representation of the constructed graphs by adding additional node and edge features beyond simple relations. Furthermore, we aim to extend the model to support multiple ontologies, enabling it to better support annotation tasks in multi-domain scenarios.

Acknowledgments

The authors would like to thank the German Federal Government, the German State Governments, and the Joint Science Conference (GWK) for their funding and support as part of the NFDI4Energy consortium. The work was funded by the German Research Foundation (DFG) – 501865131 within the German National Research Data Infrastructure (NFDI, www.nfdi.de).

This work is supported by the Helmholtz Association Initiative and Networking Fund on the HAICORE@KIT partition and the Helmholtz Metadata Collaboration (HMC).

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-4o and Grammarly for: Grammar and spelling checks. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al., The fair guiding principles for scientific data management and stewardship, Scientific data 3 (2016) 1–9.

[2] J. Tu, J. Fan, N. Tang, P. Wang, G. Li, X. Du, X. Jia, S. Gao, Unicorn: A unified multi-tasking model for supporting matching tasks in data integration, Proceedings of the ACM on Management of Data 1 (2023) 1–26.

[3] E. Rahm, P. A. Bernstein, A survey of approaches to automatic schema matching, the VLDB Journal 10 (2001) 334–350.

[4] Z. Bellahsene, A. Bonifati, F. Duchateau, Y. Velegrakis, On evaluating schema matching and mapping, in: Schema matching and mapping, Springer, 2010, pp. 253–291.

[5] D. Aumueller, H.-H. Do, S. Massmann, E. Rahm, Schema and ontology matching with coma++, in: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, 2005, pp. 906–908.

[6] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, arXiv preprint arXiv:1908.10084 (2019).

[7] Y. Suhara, J. Li, Y. Li, D. Zhang, Ç. Demiralp, C. Chen, W.-C. Tan, Annotating columns with pre-trained language models, in: Proceedings of the 2022 International Conference on Management of Data, 2022, pp. 1493–1503.

[8] Z. Tan, D. Li, S. Wang, A. Beigi, B. Jiang, A. Bhattacharjee, M. Karami, J. Li, L. Cheng, H. Liu, Large language models for data annotation and synthesis: A survey, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 930–957.

[9] M. Parciak, B. Vandevoort, F. Neven, L. M. Peeters, S. Vansummeren, Llm-matcher: A name-based schema matching tool using large language models, in: Companion of the 2025 International Conference on Management of Data, 2025, pp. 203–206.

[10] J. Freire, G. Fan, B. Feuer, C. Koutras, Y. Liu, E. Peña, A. S. Santos, C. T. Silva, E. Wu, Large language models for data discovery and integration: Challenges and opportunities., IEEE Data Eng. Bull. 49 (2025) 3–31.

[11] J. Dong, B. Fatemi, B. Perozzi, L. F. Yang, A. Tsitsulin, Don't forget to connect! improving rag with graph-based reranking, arXiv preprint arXiv:2405.18414 (2024).
[12] M. S. Zaoad, N. Zawad, P. Ranade, R. Krogman, L. Khan, J. Holt, Graph-based re-ranking: Emerging techniques, limitations, and opportunities, arXiv preprint arXiv:2503.14802 (2025).
[13] H. Zhu, D. Xu, Y. Huang, Z. Jin, W. Ding, J. Tong, G. Chong, Graph structure enhanced pre-training language model for knowledge graph completion, IEEE Transactions on Emerging Topics in Computational Intelligence 8 (2024) 2697–2708.
[14] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, H. Li, Learning to rank: From pairwise approach to listwise approach, volume 227, 2007, pp. 129–136. doi:10.1145/1273496.1273513.
[15] K. Reed, H. {Tayyar Madabushi}, Faster bert-based re-ranking through candidate passage extraction, in: The Twenty-Ninth Text REtrieval Conference (TREC 2020), 2020, pp. 1–5.
[16] A. G. D. Francesco, C. Giannetti, N. Tonellotto, F. Silvestri, Graph neural re-ranking via corpus graph, 2024. URL: https://arxiv.org/abs/2406.11720. arXiv:2406.11720.
[17] J. Luo, X. Chen, B. He, L. Sun, Prp-graph: Pairwise ranking prompting to llms with graph aggregation for effective text re-ranking, 2024, pp. 5766–5776. doi:10.18653/v1/2024.acl-long.313.
[18] L. Gienapp, M. Fröbe, M. Hagen, M. Potthast, Sparse pairwise re-ranking with pre-trained transformers, in: Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR '22, ACM, 2022, p. 72–80. URL: http://dx.doi.org/10.1145/3539813.3545140. doi:10.1145/3539813.3545140.
[19] M. Rathee, S. MacAvaney, A. Anand, Guiding retrieval using llm-based listwise rankers, 2025. URL: https://arxiv.org/abs/2501.09186. arXiv:2501.09186.
[20] S. Brody, U. Alon, E. Yahav, How attentive are graph attention networks?, arXiv preprint arXiv:2105.14491 (2021).
[21] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G. Hullender, Learning to rank using gradient descent, in: Proceedings of the 22nd international conference on Machine learning, 2005, pp. 89–96.
[22] Y.-M. Tamm, R. Damdinov, A. Vasilev, Quality metrics in recommender systems: Do we calculate metrics consistently?, in: Proceedings of the 15th ACM conference on recommender systems, 2021, pp. 708–713.