DBLPLink 2.0 - An Entity Linker for the DBLP Scholarly Knowledge Graph

Debayan Banerjee[1,*], Tilahun Abedissa Taffa[1,2] and Ricardo Usbeck[1]

[1]Leuphana University of Lüneburg, Lüneburg, Germany
[2]University of Hamburg, Hamburg, Germany

Abstract

In this work we present an entity linker for DBLP's 2025 version of RDF-based Knowledge Graph. Compared to the 2022 version, DBLP now considers publication venues as a new entity type called dblp:Stream. In the earlier version of DBLPLink, we trained KG-embeddings and re-rankers on a dataset to produce entity linkings. In contrast, in this work, we develop a zero-shot entity linker using LLMs using a novel method, where we re-rank candidate entities based on the log-probabilities of the "yes" token output at the penultimate layer of the LLM. The demo can be accessed at https://dblplink-2.skynet.coypu.org/.

Keywords

Entity Linker, DBLP, Knowledge Graphs, LLM

1. Introduction and Related Work

Entity Linking (EL) is a task in natural language processing (NLP) that involves mapping named entities mentioned in text to their unique identifiers in a knowledge graph (KG). For example, in the question "Where did Albert Einstein study?", the label "Albert Einstein" needs to be linked to the unique entity identifier Q937[1] in the Wikidata KG [1]. Various entity linking systems have been developed [2] for general-purpose KGs like Wikidata, as well as for specialized domains such as biomedical [3] or financial [4] knowledge graphs.

Scholarly KGs are a specific type of knowledge graph focused on bibliographic data related to academic publications, authors, institutions etc. Examples of well-known scholarly KGs include OpenAlex [5], ORKG [6], and DBLP [7]. In this work, we concentrate on the DBLP KG, which is specifically designed for the computer science domain and is consequently smaller than more comprehensive scholarly KGs. DBLP—originally short for Data Bases and Logic Programming—was created in 1993 by Michael Ley at the University of Trier, Germany [5].

For entity linking over DBLP, a system named Deola [8] was able to link author entities to DBLP documents in 2016. Notably, this predated the availability of DBLP in th RDF format. In 2022, we released DBLPLink [9] (which we henceforth refer to as DBLPLink 1.0), an entity linker built for the initially released version of the DBLP KG. The DBLP KG schemas before 2024 were built primarily

ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan
*Corresponding author.
debayan.banerjee@leuphana.de (D. Banerjee); tilahun.taffa@leuphana.de (T. A. Taffa); ricardo.usbeck@leuphana.de (R. Usbeck)
0000-0001-7626-8888 (D. Banerjee); 0000-0002-2476-8335 (T. A. Taffa); 0000-0002-0191-7211 (R. Usbeck)

[1]https://www.wikidata.org/wiki/Q937

around two major entity types: Creator and Publication. Subsequently in June 2024, DBLP introduced[2] a new entity type dblp:Stream which encompasses multiple sub-classes under the broad category of publication venues, for example, conferences, journals, series and repositories.

Our initial thought was to retrain DBLPLink 1.0 on the new KG and produce DBLPLink 2.0. However, moving DBLPLink 1.0 to a new KG requires computing new KG embeddings for all entities, retraining the entity label span detector, and re-training the re-ranker. In light of recent approaches using LLM-based prompting and zero-shot methods, we decided to build a new architecture from scratch for DBLPLink 2.0. DBLPLink 2.0 is able to link Person and Publication entity types as before, and additionally, can also link Stream entity types. DBLPLink 2.0 can be accessed at https://dblplink-2.skynet.coypu.org/. The code and data used to build this demo can be accessed at https://github.com/semantic-systems/dblplink-2.0.

2. User Interface

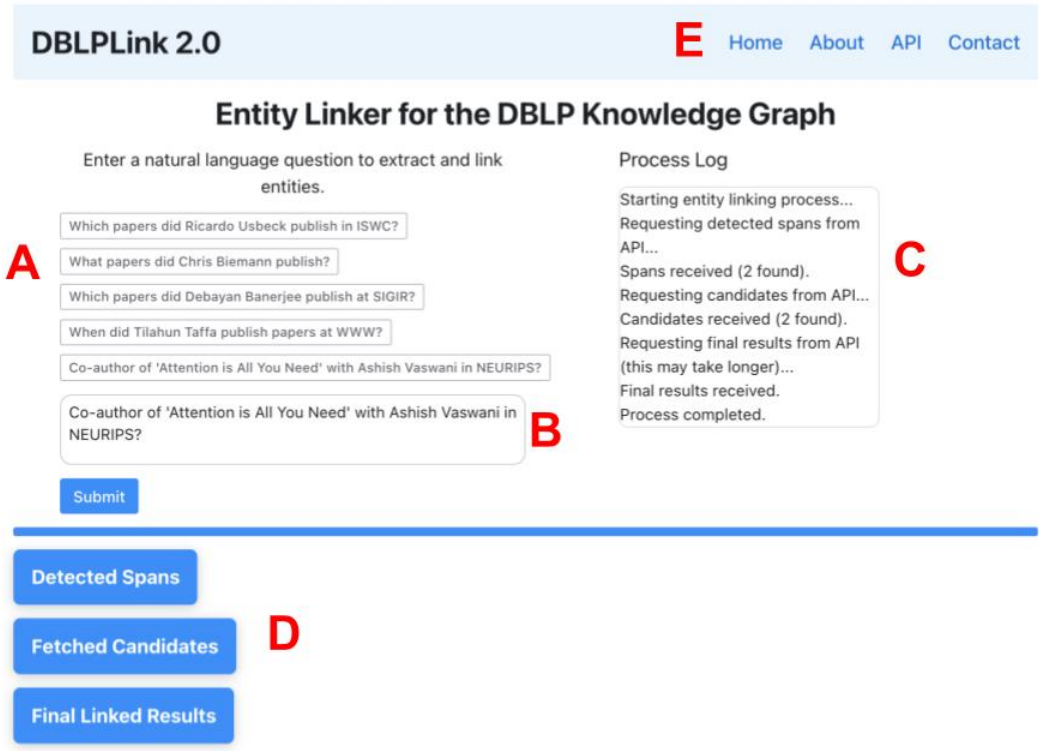

Figure 1: The DBLPLink 2.0 Web Interface

As seen in Figure 1, the web UI is divided into five elements. **A** presents a set of question templates which maybe clicked and selected to fill up the text box. **B** carries the text box where the user may type an input text, and click on Submit to start the entity linking process. **C** displays a process log which is dynamically updated from the backend, keeping the user informed on the current step being executed. **D** is results area, where the detected mention spans and their types are displayed. Later, the fetched candidates form the text search are displayed. Finally the linked results are displayed under

[2]https://blog.dblp.org/2024/06/14/the-dblp-knowledge-graph-major-extension-and-an-update-to-the-rdf-schema/

"Final Linked Results". **E** is a carousel of sub-pages, which provides further information, such as how to access the entity linker via an API call, more information about the backend entity linker architecture, and details of how to contact the authors and maintainers.

Further, as seen in Figure 2, the final linked results tab, when expanded, displays a sorted list of linked entities by log probability score, per span. The **first column** is the Span ID, where 0 stands for the first span, 1 stands for the second span and so forth. The **second column** is the entity label of the candidate as fetched from the Elasticsearch label database. The **third column** displays the DBLP type for the entity candidate. The **fourth column** displays the log probability score of the given entity. Note that the scores are in negative, and hence, they appear sorted in descending absolute value scores. The **fifth column** is also called the evidence sentence, which is the triple that produced the strongest log probability score for among all the triples for this given entity. The **sixth column** provides a clickable URL link for the entity, which takes the user to the entity's DBLP page.

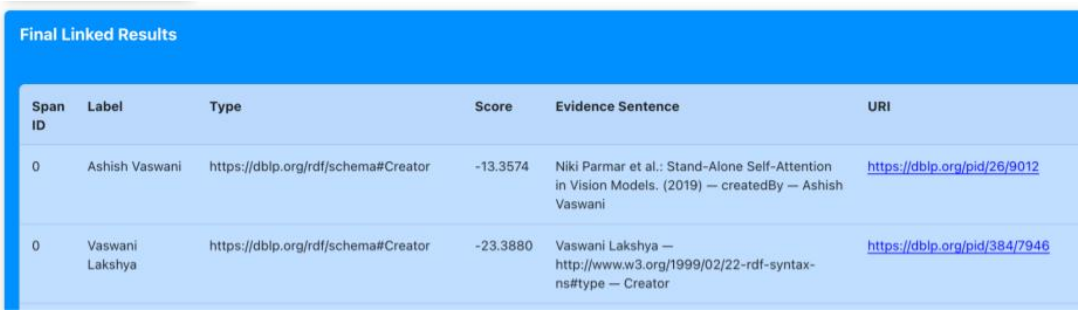

Figure 2: The "Final Linked Results" Tab when expanded

3. Entity Linker Architecture

Our entity linking pipeline combines prompted large language models (LLMs), type-specific retrieval from an Elasticsearch index, and neighborhood-based re-ranking using KG context. We illustrate the method using the input question:

```
Who are the co-authors of Ashish Vaswani in "Attention is All You
Need" in neurips?
```

3.1. Mention and Type Extraction via Prompted LLM

We first extract named entity mentions from the input using a prompted LLM. The prompt is as follows:

```
 You are an information extraction assistant.
Extract named entities from the following sentence and classify them into
one of the following types: person, publication, venue.
Let the output be a JSON array of objects with fields 'label' and 'type'.
Not all types may be present in a sentence. Now extract entities from the
following sentence:
Sentence: "Who are the co-authors of Ashish Vaswani in the 'attention is
all you need' paper in neurips?"
Entities:
```

The LLM produces:

```
[
  {"label": "Ashish Vaswani", "type": "person"},
  {"label": "attention is all you need", "type": "publication"},
  {"label": "neurips", "type": "venue"}
]
```

3.2. Candidate Entity Retrieval

Each extracted label is matched against a type-specific Elasticsearch index to retrieve a list of candidate entities. For example:

- `"Ashish Vaswani"` → [Ashish Vaswani, Vicky Vaswani, ...]
- `"attention is all you need"` → [doi:10.5555/attention-paper, ...
- `"neurips"` → [NeurIPS, NeurIPS 2022, NeurIPS 2023, ...]

3.3. Knowledge Graph Neighborhood Expansion

For each candidate entity, we fetch up to N one-hop neighbors from a knowledge graph. These triples are converted into readable sequences using a template of the form:

```
[Head] - [Relation] - [Tail]
```

Example for `Ashish Vaswani (author)`:

- `Ashish Vaswani - authored - attention is all you need`
- `Ashish Vaswani - affiliated with - Google Brain`
- `Ashish Vaswani - published at - NeurIPS`

This yields a set of short sentences describing the local graph structure of each candidate.

3.4. Candidate Scoring with LLM Log-Probability

Each linearized triple is evaluated by an LLM in the context of the original question. The prompt is:

```
Given this input text: "Who are the co-authors of Ashish Vaswani in the
'attention is all you need' paper in neurips?"
And the neighborhood context:
Ashish Vaswani - authored - attention is all you need
Is this the correct entity?
Answer with 'yes' or 'no'.
```

We extract the log-probability of the next token being `"yes"` (before generation), which serves as a soft alignment score for that triple. Each candidate entity receives multiple such scores — one per triple. These are aggregated using mean pooling, where the average log-probability over all triples is computed.

3.5. Entity Re-ranking

All candidate entities for a given mention are ranked according to their aggregated log-probability scores. The top-ranked candidate is selected as the final linked entity.

4. Implementation Details

The web demo is implemented using the Reflex web development framework[3] which allows building dynamic web interfaces written purely in Python. For finding optimal parameters for the different components of the entity linker pipeline, we randomly selected a set of 100 questions from the test set of the DBLP_QuAD dataset [10]. As seen in Table 1, we tested several different LLMs of small sizes, keeping in mind the limited GPU infrastructure available to us as university based researchers. We tested *0.5B, 1.5B, 3B, 7B, 14B* models of the *Qwen-2.5* family and *Llama-3.1-8B* and *Mistral-7B-Instruct-v0.2*. Based on the results of our experiments, we found the Mistral model lagging far behind, with F1 score of 0.09. In comparison, Qwen-2.5-3B provided an optimal balance between size and performance, hence the web demo makes use of this model. The "text only" performance in the fourth row is a setting where the top text-based match is chosen as the final entity linking result. In effect, the subsequent neighbourhood-based re-ranking step is skipped. When comparing this result to the row above, it is clear that the entity linker is performing better than pure text-match-based entity linking. Additionally, from the last column's results, it seems that only for 62% of the cases do the labels produced by the mention span detector translate to relevant candidates being fetched from the Elasticsearch labels database. All the experiments were performed with a setting of n=10 and k=10, where n=number of candidates from text search and k=number of neighbours from entities. We performed experiments with greater n and k, but saw negligible improvements when compared to the rise in execution time given the larger context to be parsed by the LLMs. Hence, we settled for values of 10 for n and k.

5. Limitations and Future Work

Due to non-availability of a new entity linking dataset over the current DBLP schema, we were unable to perform extensive evaluation for this task, especially on the new dblp:Stream entity type. Also, because the underlying KGs are different, we could not directly compare DBLPLink 2.0's performance with DBLPLink 1.0. As future, work, we shall prioritise the collection of a new dataset which would allow deeper analysis of our entity linker.

Table 1
Evaluation Results on a test set of 100 questions from DBLP_QuAD

Model / Setting	F1	MRR	Hits@1	Hits@5	Hits@10
Qwen-0.5b	0.0000	0.0000	0.0000	0.0000	0.0000
Qwen-1.5b	0.2433	0.3721	0.3100	0.4400	0.4600
Qwen-3b	**0.4400**	0.5388	**0.4900**	0.5900	0.6200
Qwen-3b text only	0.3867	0.4844	0.4300	0.5600	0.6200
Qwen-14b	0.4300	**0.5525**	0.5000	**0.6200**	0.6200
Qwen-14b	0.0233	0.1022	0.0300	0.0900	0.6200
Qwen-14b	0.4200	0.5274	0.4600	0.6100	0.6200
LLaMA-3.1-8b	0.4000	0.4922	0.4000	0.6200	0.6400
Mistral-7b	0.0900	0.1841	0.1000	0.2600	0.4800

[3] https://reflex.dev/

6. Declaration on the Use of Generative AI

No use of generative AI was made in writing this paper. We relied on the spell-check feature of Sharelatex software which was provided to us by the University of Leuphana as a tool to write research papers. ChatGPT was used for generating the initial templates of the code that the demo runs on. The code was later improved by the authors themselves to make it fully functional.

References

[1] D. Vrandečić, M. Krötzsch, Wikidata: A Free Collaborative Knowledgebase, Communications of the ACM 57 (2014) 78–85. URL: https://dl.acm.org/doi/10.1145/2629489.

[2] Ö. Sevgili, A. Shelmanov , M. Arkhipov, A. Panchenko, C. Biemann, Neural Entity Linking: A Survey of Models based on Deep Learning, Semantic Web Journal 13 (2022) 527–570. URL: https://dl.acm.org/doi/10.3233/SW-222986.

[3] E. French, B. T. McInnes, An Overview of Biomedical Entity Linking throughout the Years, Journal of Biomedical Informatics 137 (2023) 104–252. URL: https://www.sciencedirect.com/science/article/abs/pii/S153204642200257X.

[4] S. Elhammadi, L. V.S. Lakshmanan, R. Ng, M. Simpson, B. Huai, Z. Wang, L. Wang, A High Precision Pipeline for Financial Knowledge Graph Construction, in: Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 967–977. URL: https://aclanthology.org/2020.coling-main.84.

[5] J. Priem, H. Piwowar, R. Orr, Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts, 2022. URL: https://arxiv.org/abs/2205.01833. arXiv:2205.01833.

[6] M. Stocker, A. Oelen, M. Y. Jaradeh, M. Haris, O. A. Oghli, G. Heidari, H. Hussein, A.-L. Lorenz, S. Kabenamualu, K. E. Farfar, M. Prinz, O. Karras, J. D'Souza, L. Vogt, S. Auer, Fair scientific information with the open research knowledge graph, FAIR Connect 1 (2023) 19–21. URL: https://journals.sagepub.com/doi/abs/10.3233/FC-221513. doi:10.3233/FC-221513. arXiv:https://journals.sagepub.com/doi/pdf/10.3233/FC-221513.

[7] M. Ley, The dblp computer science bibliography: Evolution, research issues, perspectives, in: A. H. F. Laender, A. L. Oliveira (Eds.), String Processing and Information Retrieval, SpringerLink Bücher, Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 1–10. doi:10.1007/3-540-45735-6{\textunderscore}1.

[8] Y. Liu, W. Shen, X. Yuan, Deola: A system for linking author entities in web document with dblp, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 2449–2452. URL: https://doi.org/10.1145/2983323.2983330. doi:10.1145/2983323.2983330.

[9] D. Banerjee, Arefa, R. Usbeck, C. Biemann, DBLPLink: An Entity Linker for the DBLP Scholarly Knowledge Graph, in: Proceedings of the 22nd International Semantic Web Conference Posters, Demos and Industry Tracks, volume 3632, Athens, Greece, 2023. URL: https://ceur-ws.org/Vol-3632/ISWC2023_paper_428.pdf.

[10] D. Banerjee, S. Awale, R. Usbeck, C. Biemann, Dblp-quad: A question answering dataset over the dblp scholarly knowledge graph, in: Proceedings of the 13th International Workshop on Bibliometric-enhanced Information Retrieval co-located with 45th European Conference on Information Retrieval (ECIR 2023), Dublin, Ireland, April 2nd, 2023, pp. 37–51. URL: https://ceur-ws.org/Vol-3617/paper-05.pdf.