DBLPLink: An Entity Linker for the DBLP Scholarly Knowledge Graph

Debayan Banerjee[1], Arefa[2], Ricardo Usbeck[1] and Chris Biemann[1]

[1]Universität Hamburg, Hamburg, Germany
[2]Jamia Milia Islamia, New Delhi, India

Abstract

In this work, we present a web application named DBLPLink, which performs entity linking over the DBLP scholarly knowledge graph. DBLPLink uses text-to-text pre-trained language models, such as T5, to produce entity label spans from an input text question. Entity candidates are fetched from a database based on the labels, and an entity re-ranker sorts them based on entity embeddings, such as TransE, DistMult and ComplEx. The results are displayed so that users may compare and contrast the results between T5-small, T5-base and the different KG embeddings used. The demo can be accessed at https://ltdemos.informatik.uni-hamburg.de/dblplink/. Code and data shall be made available at https://github.com/uhh-lt/dblplink.

1. Introduction and Related Work

Entity Linking (EL) is a natural language processing (NLP) task that involves associating named entities mentioned in text to their corresponding unique identifiers in a knowledge graph (KG). For example, in the question: *Who is the president of USA?*, the named entity span of *USA* has to be linked to the unique identifier Q30[1] in the Wikidata KG [1]. Several entity linkers exist [2] over general purpose KGs such as Wikidata, and more specialized KGs, such as bio-medical [3] or financial KGs [4], however, to the best of our knowledge, no working entity linker exists for scholarly KGs.

A scholarly KG is a special sub-class of KGs, which contains bibliographic information about research publications, authors, institutions etc. Some well-known scholarly KGs are the OpenAlex[2], ORKG[3] and DBLP[4]. In this work, we focus on the DBLP KG, which caters specifically to computer science, and as a result, is smaller in size than other scholarly KGs. DBLP, which used to stand for Data Bases and Logic Programming[5], was created in 1993 by Michael Ley at the University of Trier, Germany [5]. At the time of its release[6], the RDF dump consisted of 2,941,316 person entities, 6,010,605 publication entities, and 252,573,199 RDF triples.

ISWC 2023 Posters and Demos: 22nd International Semantic Web Conference, November 6–10, 2023, Athens, Greece
debayan.banerjee@uni-hamburg.de (D. Banerjee); arefa.muzaffar@gmail.com (Arefa); ricardo.usbeck@uni-hamburg.de (R. Usbeck); chris.biemann@uni-hamburg.de (C. Biemann)

[1]https://www.wikidata.org/wiki/Q30
[2]http://openalex.org/
[3]https://orkg.org/
[4]https://dblp.org/
[5]https://en.wikipedia.org/wiki/DBLP
[6]https://blog.dblp.org/2022/03/02/dblp-in-rdf/

DBLPLink can handle simple and complex questions pertaining to authorship, venues, institutions and other information available in the DBLP KG.

2. Web Interface

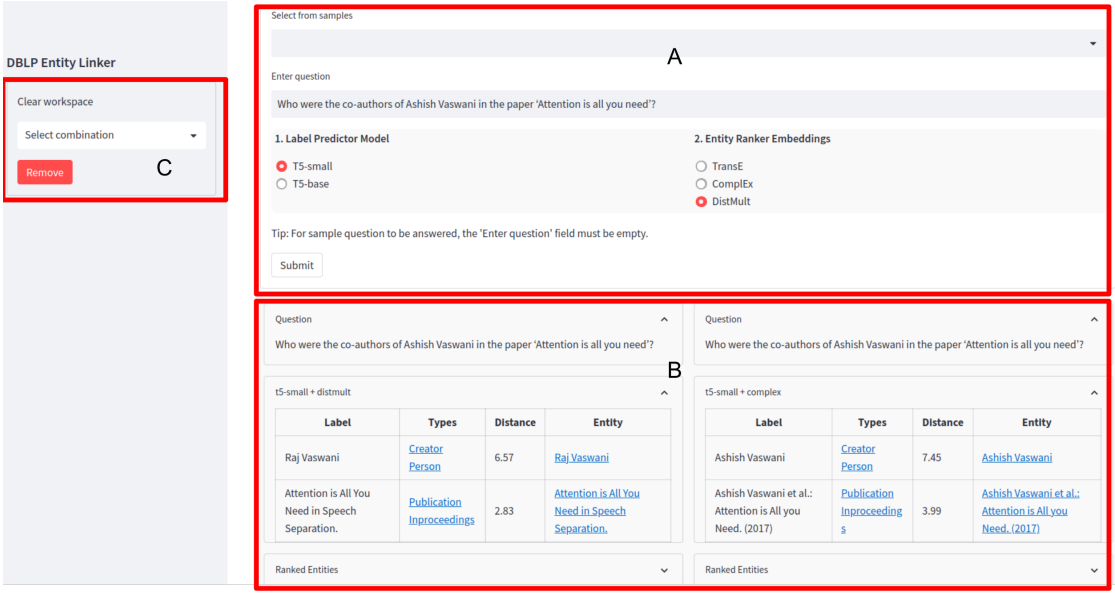

Figure 1: User interface of DBLPLink. The question reads: "Who were the co-authors of Ashish Vaswani in the paper 'Attention is all you need'?"

As shown in Figure 1, the UI consists of three main parts. In **Section A**, the user can either type a question as input or select a question from the drop-down menu. Further, the user can select which model to use for label span detection, and which embeddings to use for re-ranking of entities. In **Section B**, the results of DBLPLink are displayed. First, the top-ranked entity for each detected span is displayed, with a corresponding label and type from the DBLP KG. A hyperlink to the entity, which points to the original DBLP entity web page is also shown. Additionally, a `distance` metric is shown which denotes how close a match this entity is to the input question. A lower distance means a better match. Towards the bottom of the UI, we can briefly see collapsible boxes called "Ranked Entities", which further display the top 10 ranked entities for each of the detected label spans. Lastly, in **Section C**, the user has an option to remove certain combinations of results from the screen, if the UI becomes too cluttered. Our expectation is that the user shall try multiple combinations of T5 and entity embeddings to compare and contrast the results, which may need occasional cleanup from the UI.

3. Architecture

3.1. Label and Type Generation

As seen in Figure 2, the first step is to produce salient labels and types from the given input question. For this purpose, we use the DBLP-QuAD [6] dataset to fine-tune T5-small and T5-base [7] models, on the task of producing entity labels and types from the input question.

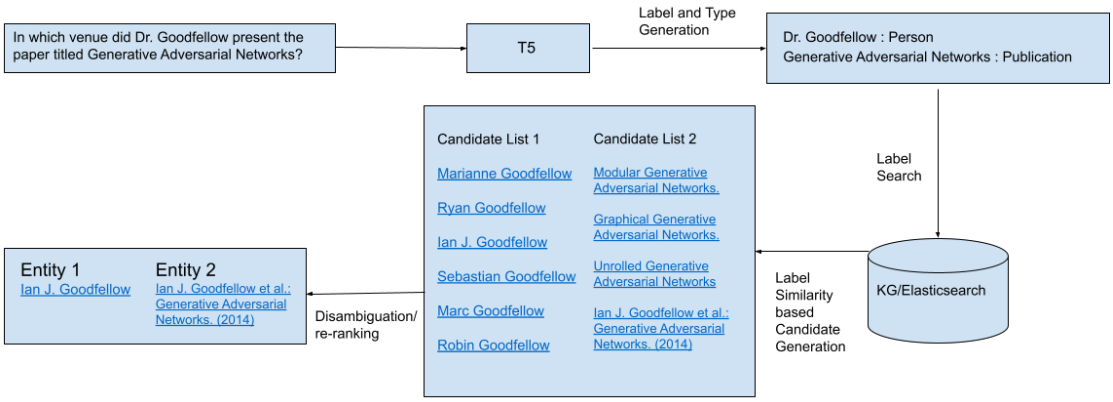

Figure 2: Architecture of DBLPLink.

3.2. Candidate Generation

With the entity labels and types produced in the previous step, a free-text-search is performed on an Elasticsearch[7] instance, which contains entity URLs with their corresponding labels. The results are further filtered by the types. This gives us a list of candidate entities. In normal operation of the demo application, we present the top-ranked candidate as the final linked entity. We only proceed to the disambiguation stage if the top entity candidate has a label, that is the same as another entity in the candidate list.

3.3. Disambiguation

In case two entities in the candidate list share the same label, we proceed with disambiguation, which requires a further re-ranking of the candidate list. For this, we follow a common approach of using Siamese neural networks [8] for learning text similarity between text pairs [9]. We embed the input question and the candidate entities in a common embedding space. For this purpose, we create a 969-dimensional embedding, where for a given question, we use the first 768 dimensions for the BERT embedding. We fill the remaining 201 dimensions with zeros. For the entity candidates, we fill the first 768 dimensions with the BERT embedding of the entity label, while the next 200 dimensions are reserved for the entity embeddings. We use three different kinds of embeddings in our experiments, namely TransE [10], ComplEx [11], and DistMult [12]. For the remaining 969th dimension, we store the degree of string similarity match between the entity label and the input question. For training, pairs of positive and negative samples are used with a triplet ranking loss function and L2 distance metric.

During inference, a question and an entity candidate are vectorised and passed through the trained Siamese network. The cosine distance between the two resulting embeddings is computed, and the pair with the lowest distance is considered the most suitable match.

4. Evaluation

We evaluate our entity linker on the 2.000 questions of the test split of the DBLP-QuAD dataset and measure the F1-score. In Table 1, under the heading ‘Label Sorting‘, we consider the

[7]https://www.elastic.co/

		conditional-disambiguation			hard-disambiguation		
	Label Sorting	TransE	ComplEx	DistMult	TransE	ComplEx	DistMult
T5-small	0.698	0.700	0.692	0.699	0.511	0.482	0.537
T5-base	0.698	**0.701**	0.692	**0.701**	0.521	0.484	0.547

Table 1
F1-scores for the entity linking task across different combinations of span detector and entity re-ranker

top-ranked candidate after the label sorting phase as the linked entity. We perform no further disambiguation. Under the 'conditional-disambiguation' setting, we perform disambiguation only if two entities in the candidate list share the same label. Under the 'hard-disambiguation' setting, re-ranking based on Siamese network cosine distances is always run after the candidate generation phase, essentially ignoring the label sorting order.

We see that hard-disambiguation lags behind significantly in performance when compared to plain label sorting, which points to the learning that for DBLP KG, degree of string match of an author or a publication is more important than the KG embeddings. Based on this finding, we allow the web application to run in 'conditional-disambiguation' mode for better performance. In the case of conditional disambiguation, performance is marginally better when using TransE and DistMult when compared to label sorting, because not many cases of ambiguous labels exist in the DBLP-QuAD test set. However, it is evident from the hard disambiguation case, that DistMult performs the best on a pure disambiguation task. This may be explained by the inherent suitability of DistMult for 1-to-N relationships, which is close to the nature of the DBLP KG model, where one author may have several papers. On the contrary, TransE expects 1-to-1 relationships, while ComplEx works better for symmetric relationships. Another interesting outcome of the experiments is that the difference in parameter sizes of T5-small and T5-base does not produce any difference in performance. This may be explained by the fact that in the span label production task, much of the focus is on copying the right part of the input to the output. Since the learned knowledge of the model weights from the pre-training task is not being exploited, the larger size of T5-base does not seem to matter.

5. Conclusion

In this work, we presented DBLPLink, which is a web-based demonstration of an entity linker over the DBLP scholarly KG. In the future, we would like to add further interactivity to the UI where users can provide feedback on quality of the results. Additionally, a conversational interface for question answering would be desirable for question answering tasks, and we would like to build it in a future version.

6. Acknowledgements

This research is performed as a part of the ARDIAS project, funded by the "Idea and Venture Fund" research grant by Universität Hamburg, which is part of the Excellence Strategy of the Federal and State Governments. This work has additionally received funding through the German Research Foundation (DFG) project NFDI4DS (no. 460234259).

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] M. Ley, The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives, in: String Processing and Information Retrieval, volume 2476, Berlin, Heidelberg, 2002, pp. 1–10. URL: https://link.springer.com/chapter/10.1007/3-540-45735-6_1.

[6] D. Banerjee, S. Awale, R. Usbeck, C. Biemann, DBLP-QuAD: A Question Answering Dataset over the DBLP Scholarly Knowledge Graph, 2023. arXiv:2303.13351.

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020).

[8] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature Verification Using a Siamese Time Delay Neural Network, in: Advances in Neural Information Processing Systems 6, 7th NIPS Conference, Denver, Colorado, USA, 1993, pp. 737–744. URL: http://papers.nips.cc/paper/769-signature-verification-using-a-siamese-time-delay-neural-network.

[9] T. Ranasinghe, C. Orasan, R. Mitkov, Semantic Textual Similarity with Siamese Neural Networks, in: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP), Varna, Bulgaria, 2019, pp. 1004–1011. URL: https://aclanthology.org/R19-1116.

[10] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, O. Yakhnenko, Translating Embeddings for Modeling Multi-Relational Data, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, 2013, p. 2787–2795.

[11] T. Trouillon, J. Welbl, S. Riedel, G. Gaussier, E.and Bouchard, Complex Embeddings for Simple Link Prediction, in: Proceedings of The 33rd International Conference on Machine Learning, volume 48 of *Proceedings of Machine Learning Research*, New York, New York, USA, 2016, pp. 2071–2080. URL: https://proceedings.mlr.press/v48/trouillon16.html.

[12] B. Yang, W. Yih, X. He, J. Gao, L. Deng, Embedding Entities and Relations for Learning and Inference in Knowledge Bases, in: 3rd International Conference on Learning Representations, ICLR, San Diego, USA, Conference Track Proceedings, 2015. URL: http://arxiv.org/abs/1412.6575.