XSearchKG: A Platform for Explainable Keyword Search over Knowledge Graphs

Leila Feddoul[1] [0000-0001-8896-8208], Martin Birke[1] [0009-0008-2589-6532], and Sirko Schindler[2] [0000-0002-0964-4457]

[1] Heinz Nixdorf Chair for Distributed Information Systems, Friedrich Schiller University Jena, Jena, Germany
{leila.feddoul,martin.birke}@uni-jena.de
[2] Institute of Data Science, German Aerospace Center DLR, Jena, Germany
sirko.schindler@dlr.de

Abstract. One of the most user-friendly methods to search over knowledge graphs is the usage of keyword queries. They offer a simple text input that requires no technical or domain knowledge. Most existing approaches for keyword search over graph-shaped data rely on graph traversal algorithms to find connections between keywords. They mostly concentrate on achieving efficiency and effectiveness (accurate ranking), but ignore usability, visualization, and interactive result presentation. All of which offer better support to non-experienced users. Moreover, it is not sufficient to just show a raw list of results, but it is also important to explain why a specific result is proposed. This not only provides an abstract view of the capabilities and limitations of the search system, but also increases confidence and helps discover new interesting facts. We propose XSearchKG, a platform for explainable keyword search over knowledge graphs that extends our previously proposed graph traversal-based approach and complements it with an interactive user interface for results explanation and browsing.

Keywords: Explainability · Keyword Search · Knowledge Graph.

1 Introduction

Keyword search is a familiar method for information retrieval that can also be adapted to enable access to knowledge graphs (KGs). Using this common search paradigm is intuitive and especially suitable for end users. It allows to take advantage of the graph nature of the data by exploring it and thus discovering prominent associations and deriving deductions. Moreover, it does not require any technical expertise (e.g., SPARQL [1]) or knowledge of the underlying data structure. Recently, various approaches were developed to solve the problem of keyword search over KGs [17,15,4,9,14,6]. Most of those systems aim at efficiently finding and ranking subgraph-shaped structures that connect the graph elements corresponding to the keywords. However, only very few systems propose a user interface for result presentation [2,8,13,10]. This reduces the suitability for end users. Furthermore, none of the systems aims at explaining why a specific item appears among the search results in a detailed way. We observe in some cases only the usage of basic features such as the highlighting of keyword mentions in the search result snippets [10,2]. Recently, the concept of explainability has gained importance especially in the field of artificial intelligence (AI), where the aim is to support understanding the predictions performed by machine learning models [5], in order to increase user trust and causality [11]. However, this is not only relevant to AI, but also other use cases such as recommendation systems or information retrieval in general. In the context of keyword search over KGs, we define the explainability of a system as its capability to answer the following questions: *Why is a specific result relevant to the query?* and *Why is a specific result more or less important compared to others?*.

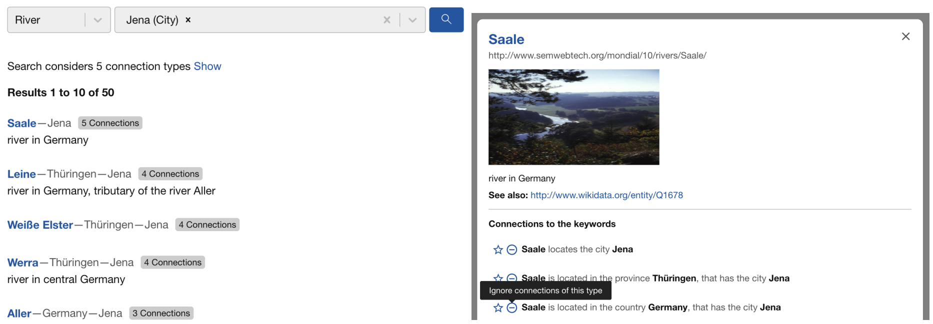
In this demo paper, we present XSearchKG, a full-stack web application for explainable keyword search over KGs. The latter is based on our approach proposed in [6,7][3], where first, the off-line phase consists of summarizing the input KG and assigning query-independent costs to its edges. The second (runtime) phase receives user keywords and a target (types of entities to be retrieved) as input, finds the nodes corresponding to the keywords and target in the summary graph, explores the graph to find subgraphs connecting keywords and target, ranks the subgraph templates (schema-level), and generates SPARQL queries corresponding to the graphs. We extend the pipeline by combining the results generated by executing the top-k SPARQL queries and propose a ranking of retrieved entities based on the scores of their subgraph templates. The implemented features to support explainability are detailed in Section 2. These are mainly based on giving the user more transparency about the generated results leveraging the connections in the schema and instance levels of the graph. The proposed visualizations allow a better understanding of the results without overwhelming the user. Moreover, they enable the discovery of new insights by getting explanations for results that may appear not correct at first glance. The conducted user evaluation is shortly described in Section 3. The source code is publicly available on GitHub [3][4]. We will demonstrate the system using MONDIAL [12] as input KG. We will use queries of different complexity levels and explore some generated results. We will also go through the different explanation views and show possible user interactions (e.g., ignore specific connections).

2 XSearchKG

The application comprises two fundamental building blocks: (1) A Node.js Express server (backend) produces ranked results with essential information such as labels, descriptions, and subgraph bindings. To avoid regenerating identical data

[3] Note that the used code version uses the cosine similarity between the embeddings of the node labels as a semantic relatedness metric. The embeddings are generated using the Universal Sentence Encoder lite

[4] https://github.com/fusion-jena/XSearchKG (a link to the demo video can be found in the README of the repository)



(a) Result List View (b) Result Detailed View

Fig. 1: Screenshots of some of the system views.

and improve performance, we utilize both standard HTTP cache headers and in-memory caches. The runtime phase of [6,7] was integrated into the backend component. (2) A React app (frontend) provides a user-friendly and responsive web interface. The application requires a running SPARQL endpoint and the files representing the summary graph generated at the first off-line phase of [6,7]. More details about the architecture are provided on GitHub. In the following, we give an overview about the different platform's features:

Input Suggestion. To perform a search, the user must enter a target type corresponding to a node on the summary graph and a set of keywords representing resources on the KG. Therefore, we assist with a suggestion service that provides appropriate candidates based on the labels allowing the user to scroll through the list and get inspiration.

Result List View. The result list (Figure 1a) shows the individual results with a label and a text description. If the KG does not provide this information we try to retrieve it from semantically identical resources in other KGs[5]. Results are ranked based on the subgraph template scores, where entities with bindings from a larger number of templates will be higher ranked. The view also offers a first glimpse of how the result is related to the keywords by showing a binding of the subgraph with the highest ranking. As in all views, we display the subgraphs in the form of a tree, with the result being the root on the left side and the keywords being (typically) the leaf nodes on the right side. This provides a clear reading direction. Since we do not want to clutter the result list, we show a sparse view consisting of only the nodes of the subgraph omitting any edge labels together with the number of subgraph templates that have a binding for this result to show results' importance. For example, in Figure 1a, the user can see that the river *Saale* is directly connected to the keyword *Jena* while the top subgraphs of the other results contain an intermediate node (*Thüringen* and *Germany*, respectively).

[5] In MONDIAL, this was done by leveraging `owl:sameAs` links to Wikidata [16]

Result Detailed View. Clicking on an item in the result list opens an overlay with detailed information about the result (Figure 1b). In addition to a description and an image, this includes a list of all subgraph templates that have a binding for this result. In this view, we provide meaningful edge labels to help understand the specific connection. If a subgraph template has multiple bindings, we will indicate this and allow the user to load more example bindings. We also offer two interaction options for each subgraph: (1) If a user considers a connection to be irrelevant, it can be ignored in the result generation by clicking on the minus icon, and (2) if a connection exactly matches the user's intention, it can be used as the only subgraph template by clicking on the star icon.

Connection Types View. Above the result list (Figure 1a), we show the number of subgraph templates that are considered in the result generation. By clicking the *Show* link, the user gets a list of all subgraph templates. This representation is similar to the one in the detailed view, but instead of bindings, the node types are shown. We also offer the same types of interactions. By clicking on any of them, the user immediately sees the effect on the result set, allowing for experimentation and finding the subset of subgraph templates that best fits the intent.

3 Evaluation

We conducted a qualitative user evaluation in live sessions with 6 participants with heterogeneous backgrounds in terms of domain and technological knowledge. We observed, that the users considered the top search results to be mostly relevant and reasonably ordered. Further down in the result list, participants often questioned whether individual results were correct. With the subgraphs provided, they were able to understand the context and to decide whether or not to consider the result. By ignoring irrelevant subgraph templates in the result generation, participants were able to obtain result lists that better matched their intentions. In addition, the evaluation revealed a number of usability issues that we addressed by making mostly minor changes to the user interface, such as providing easier access to the interactions.

4 Conclusion and Future Work

We introduced XSearchKG, a platform that makes the process of keyword search over KGs more transparent to the end users and allows more interaction capabilities. We designed features with various levels of abstraction for result explanation, ranging from sparse to detailed views. The system has a modular client-server architecture that communicates with a triple store endpoint and uses different caching mechanisms. An initial user evaluation in live sessions revealed that the proposed interface is intuitive in most of the cases without prior knowledge about KGs or the used algorithm. This also helped improve some features by leveraging the feedback of the users. We plan to test the platform using larger KGs and conduct more extensive evaluations.

References

1. Sparql 1.1 query language: W3c recommendation 21 march 2013 (2013), https://www.w3.org/TR/sparql11-query/
2. Bhalotia, G., Hulgeri, A., Nakhe, C., Chakrabarti, S., Sudarshan, S.: Keyword searching and browsing in databases using BANKS. In: Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, February 26 - March 1, 2002. pp. 431–440. IEEE Computer Society (2002). https://doi.org/10.1109/ICDE.2002.994756
3. Birke, M., Feddoul, L.: fusion-jena/XSearchKG v1.0.0 (Oct 2023). https://doi.org/10.5281/zenodo.8430121
4. Dosso, D., Silvello, G.: Search text to retrieve graphs: A scalable RDF keyword-based search system. IEEE Access **8**, 14089–14111 (2020). https://doi.org/10.1109/ACCESS.2020.2966823
5. Došilović, F.K., Brčić, M., Hlupić, N.: Explainable artificial intelligence: A survey. In: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). pp. 0210–0215 (2018). https://doi.org/10.23919/MIPRO.2018.8400040
6. Feddoul, L.: Semantics-driven Keyword Search over Knowledge Graphs. In: Proceedings of the Doctoral Consortium at ISWC 2020 co-located with 19th International Semantic Web Conference (ISWC 2020), Athens, Greece, November 3rd, 2020. CEUR Workshop Proceedings, vol. 2798, pp. 17–24. CEUR-WS.org (2020), https://ceur-ws.org/Vol-2798/paper3.pdf
7. Feddoul, L.: Top-k Query generator (Semantics-driven Keyword Search over Knowledge Graphs) (Oct 2023). https://doi.org/10.5281/zenodo.8414093
8. García, G., Izquierdo, Y., Menendez, E., Dartayre, F., Casanova, M.A.: RDF keyword-based query technology meets a real-world dataset. In: Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017. pp. 656–667. OpenProceedings.org (2017). https://doi.org/10.5441/002/edbt.2017.86
9. Ghanbarpour, A., Niknafs, K., Naderi, H.: Efficient keyword search over graph-structured data based on minimal covered r-cliques. Frontiers Inf. Technol. Electron. Eng. **21**(3), 448–464 (2020). https://doi.org/10.1631/FITEE.1800133
10. Kadilierakis, G., Nikas, C., Fafalios, P., Papadakos, P., Tzitzikas, Y.: Elas4rdf: Multi-perspective triple-centered keyword search over RDF using elasticsearch. In: The Semantic Web: ESWC 2020 Satellite Events - ESWC 2020 Satellite Events, Heraklion, Crete, Greece, May 31 - June 4, 2020, Revised Selected Papers. Lecture Notes in Computer Science, vol. 12124, pp. 122–128. Springer (2020). https://doi.org/10.1007/978-3-030-62327-2_21
11. Lipton, Z.C.: The mythos of model interpretability. Commun. ACM **61**(10), 36–43 (sep 2018). https://doi.org/10.1145/3233231
12. May, W.: Information extraction and integration with Florid: The Mondial case study. Tech. Rep. 131, Universität Freiburg, Institut für Informatik (1999), available from http://dbis.informatik.uni-goettingen.de/Mondial
13. Menendez, E.S., Casanova, M.A., Leme, L.A.P.P., Boughanem, M.: Novel node importance measures to improve keyword search over RDF graphs. In: Database and Expert Systems Applications - 30th International Conference, DEXA 2019, Linz, Austria, August 26-29, 2019, Proceedings, Part II. Lecture Notes in Computer Science, vol. 11707, pp. 143–158. Springer (2019). https://doi.org/10.1007/978-3-030-27618-8_11

14. Shi, Y., Cheng, G., Kharlamov, E.: Keyword search over knowledge graphs via static and dynamic hub labelings. In: WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020. pp. 235–245. ACM / IW3C2 (2020). https://doi.org/10.1145/3366423.3380110
15. Shi, Y., Cheng, G., Tran, T., Kharlamov, E., Shen, Y.: Efficient computation of semantically cohesive subgraphs for keyword-based knowledge graph exploration. In: WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021. pp. 1410–1421. ACM / IW3C2 (2021). https://doi.org/10.1145/3442381.3449900
16. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (Sep 2014). https://doi.org/10.1145/2629489
17. Zhang, Z., Yu, J.X., Wang, G., Yuan, Y., Chen, L.: Key-core: cohesive keyword subgraph exploration in large graphs. World Wide Web **25**(2), 831–856 (2022). https://doi.org/10.1007/s11280-021-00926-y