# A Testbed for Dual-Entity Knowledge Panels

Leon Martin[1], Andreas Henrich[1]

[1]*University of Bamberg, An der Weberei 5, 96047 Bamberg, Germany*

### Abstract
Currently, web search engines reliably display knowledge panels with summarizing information only when the issued query mentions exactly one entity. That being said, queries mentioning multiple entities are relatively common. The present paper introduces a testbed for developing and evaluating dual-entity knowledge panels. The idea is to populate these novel knowledge panels with an explanation of the relationship between the entities of dual-entity queries in order to serve the users' information need. To this end, previous research showed the feasibility of finding paths in Wikidata that connect two arbitrary entities. Although such paths provide a rich foundation for elucidating the relationship between two entities, it has not yet been studied how to present them in this context with usability in mind. Hence, this paper showcases a selection of conceivable presentation formats, including graph-based visualizations and LLM-based textual approaches, to promote research in this direction.

### Keywords
Web Search Engines, Knowledge Panels, Entity Relationship Explanation, Wikidata

## 1. Introduction

In order to cater for the users' need for faster access to relevant information, modern web search engines such as Ecosia, Bing, and Startpage[1] go beyond the conventional list-based display of search results by incorporating supplementary components on their result pages. One prominent example of these components are Knowledge Panels (KPs), which are typically positioned in the top right corner of the result page. These rectangular interface elements are designed to provide concise and curated information on an entity mentioned in the query sourced from dedicated knowledge bases known as Knowledge Graphs (KGs) [1, 2]. By integrating such KPs, users can quickly access relevant details and insights about entities of interest without having to navigate the list of search results. As described in [3], current web search engines reliably display KPs only in response to single-entity queries, i.e., queries that mention exactly one entity, though. In contrast, for dual-entity queries, i.e., queries that mention exactly two entities, different behavior occurs: in some occasions, no KP is displayed at all, whereas sometimes only a KP for one of the entities is presented. In previous work [4, 3], we explained that we regard this as a missed opportunity because KPs for dual-entity queries[2], i.e., dual-entity KPs, could provide an explanation of the relationship between the two entities, thereby potentially serving the users'

[1]https://www.ecosia.org, https://www.bing.com, https://www.startpage.com (accessed 2023/09/05)

[2]In [3], we analyzed a widely recognized dataset of 10,000 queries using a state-of-the-art entity linker and found that over 10% of them mentioned at least two entities.

information need, as well. We thus proposed a bidirectional A* search algorithm for finding meaningful paths between arbitrary entities in a KG that could serve as basis for generating an explanation of the entities' relationship [4, 3]. Given the scope of web search engines, we employed Wikidata [5], an open-domain KG, to be able to handle queries mentioning entities from virtually any domain. As a follow-up to that work, the present paper also focuses on paths from Wikidata for populating KPs with explanations of entity relationships.

That being said, it has not been studied how path-based entity relationship explanations could be presented in the special context of KPs with usability in mind. As a first step in this novel line of research, the present paper introduces a testbed[3] for developing and evaluating dual-entity KPs, in particular the presentation format that is used to convey the explanation of the entity relationship to the users. To promote research in this direction, some examples of conceivable presentation formats are shown, as well.

The remainder of the present paper is structured as follows: Section 2 discusses foundations and related work. Then, Section 3 introduces the testbed itself. Section 4 demonstrates a selection of conceivable presentation formats, before Section 5 draws a conclusion.

## 2. Foundations & Related Work

Explanations of the relationship between two entities in a KG serve as the central foundation of the present paper. In [6], the task of entity relationship explanation is defined as follows:

> "Given a pair of entities $e$ and $e'$, provide an explanation, i.e., a textual description, supported by a KG, of how the pair of entities is related."

In accordance with this definition, the bidirectional A* search algorithm from [4, 3] detects paths in Wikidata that connect $e$ and $e'$ in a meaningful way. With respect to the KP context, $e$ and $e'$ correspond to the entities mentioned in a dual-entity query issued to a web search engine. The meaningfulness requirement, which is mandatory for generating entity relationship explanations that are useful for the users, is met by considering the semantic distances between entities as part of the search heuristics guiding the algorithm.

Since Wikidata is a KG that leverages the Resource Description Framework (RDF) [7], its information is encoded in the form of triples, each comprising a subject, a predicate, and an object. A predicate represents a property, which is a binary relation between the subject and the object that can be interpreted in both directions [8]. In Wikidata, the entities, which are part of the combined set of subjects and objects, use proprietary identifiers with a leading $Q$, while the predicates use proprietary identifiers with a leading $P$, in addition to the typical Internationalized Resource Identifiers (IRIs). Wikidata can therefore be interpreted as a graph $G = (V, E)$ where the vertices $V$ are the combined set of subjects and objects and the edges $E$ are the instances of predicates. Accordingly, a path $P$ between $e \in V$ and $e' \in V$ consists of a set of vertices $V_P \subseteq V$ and a set of edges $E_P \subseteq E$, thereby also qualifying as a (sub)graph. The length of a path is defined as the number of edges, i.e., one less than the number of entities, on the path [9].

---

[3]The testbed implementation is available in the GitHub repository at https://github.com/uniba-mi/dual-entity-panels (accessed 2023/09/05), which is also indexed in the Software Heritage Project's archive (https://archive.softwareheritage.org; accessed 2023/09/05).

(a) A direct path from $e$ to $e'$       (b) A direct path from $e'$ to $e$

(c) A path composed of a direct path from $e$ to an intersecting entity $v_i$ and a direct path from $e'$ to an intersecting entity $v_i$
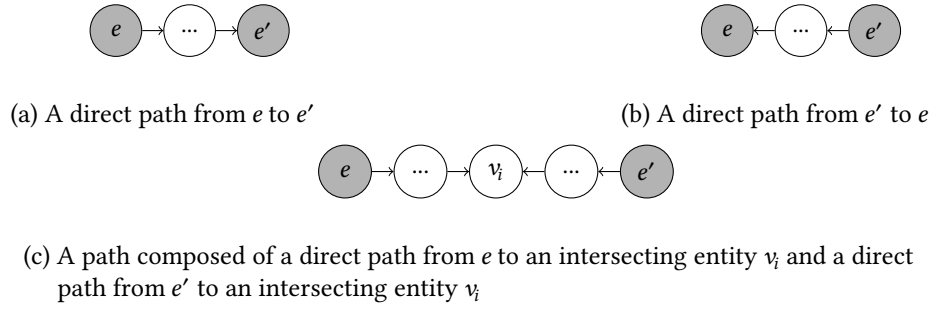
**Figure 1:** The graph patterns that can be found using the bidirectional A* search algorithm from [4], adopted from [3]. $e$ and $e'$ correspond to the entities of a dual-entity query, between which a path is searched. Nodes with ... are placeholders for series of $n \geq 0$ entities.

For exchanging RDF data, various serialization formats exist. The testbed expects paths in the Terse RDF Triple Language (Turtle) [10] as an input. Due to its characteristics, the bidirectional A* search algorithm from [4, 3] can only find paths following the patterns shown in Figure 1. That being said, the testbed is able to parse Turtle-formatted paths with arbitrary patterns even though the focus of the present paper remains on paths with the depicted patterns.

To generate a dual-entity KP based on a Turtle-formatted path, the explanation of the entity relationship encoded within the path has to be presented in some way. To this end, there are various options, some of which are fundamentally different. For instance, graph-based visualizations are just as conceivable as textual representations in natural language. Thus, we use the umbrella term *presentation formats* to subsume the range of options for presenting the entity relationship in a dual-entity KP, disregarding the utilized modalities and media types.

Regarding the presentation of paths from Wikidata in particular, it is important to consider the particularities of the Wikidata knowledge graph. Entities and properties in Wikidata feature many (contextual) information that can be leveraged to present paths in a more user-friendly manner. Most importantly, there are triples that provide natural language labels and descriptions for each entity and property. To give an example, consider the following path from Wikidata with Q7958 (explanation) representing $e$ and Q46857 (scientific method) representing $e'$[4]:

Q7958 (explanation) $-$P366 (has use)$\rightarrow$ Q352842 (teaching) $-$P31 (instance of)$\rightarrow$
Q11862829 (academic discipline) $-$P1269 (facet of)$\rightarrow$
Q336 (science) $\leftarrow$P1535 (used by)$-$ Q46857 (scientific method)

Using the testbed that will be introduced in Section 3, one can implement dual-entity KPs that leverage the presentation format from the example and evaluate it using standard methods from human-computer interaction, thereby deepening the understanding of dual-entity KP usability. In addition to labels and descriptions, there are many other properties that could be leveraged for presenting the entity relationships. The testbed imposes no restrictions on the use of supplementary information as long as it is encoded in the Turtle format.

The visualization of graphs has been studied intensively in the past [11, 12]. Occasionally, one can find side notes on how individual paths in a graph visualization could be highlighted.

---

[4]In this notation adopted from [3], the properties within the arrows (edges) connect the surrounding entities (nodes) in the respective direction. For both properties and entities, the Wikidata IDs and the labels are provided.

However, the presentation of particular paths representing entity relationships in the context of KPs has not been investigated so far. Nevertheless, insights from previous work on graph visualization should be applicable to path visualization as well, since paths themselves are, by definition, graphs, albeit simple examples of them. Disregarding the particular presentation format that is employed, fundamental principles of interface design still apply, as well. This includes, for instance, Nielsen's usability heuristics [13] but also more modern guidelines like the visual design principles postulated in [14].

Recent advances in machine learning and more specifically natural language generation enable the usage of textual presentation formats that were difficult to implement before. Previous work considered the task of natural language generation as a compound problem comprising several independent tasks that address some aspect of the generation process like the text structuring or the linguistic realization [15]. Despite significant effort, the resulting texts often lack quality. Representing the current state-of-the-art, Large Language Models (LLMs) such as GPT-4 [16] leverage a holistic approach for the natural language generation problem to generate high-quality texts. ChatGPT, the conversational AI based on GPT, allows users to easily issue specific requests via prompts. For instance, one can request a textual description of certain (semi-)structured data. For our use case, this ability can be utilized to implement a presentation format that uses an actual natural language explanation to convey the entity relationship, just as prescribed in the definition of entity relationship explanation from [6].

## 3. The Testbed

The testbed is implemented as a web application using Svelte[5] and is Docker-ized[6] for ease of use and reproducibility. As depicted in Figure 2, the user interface features a central input group allowing the selection of the two entities of a dual-entity query[7]. After the selection, users can press a button to trigger the generation of the corresponding dual-entity KP. Disregarding the presentation format, each dual-entity KP comprises the labels, the IDs, and the descriptions of the two entities. Below that, there is a dropdown menu for selecting one of the available presentation formats. By default, the presentation format called *Turtle* is selected, which presents the path as the raw Turtle document. To add other presentation formats, one only has to add a Svelte component containing the corresponding code[8].

Note that the testbed is currently not connected to a pathfinding backend since its purpose is to facilitate the implementation and evaluation of presentation formats in an isolated manner. The paths that can be selected via the central input group thus originate from Turtle-formatted paths that are hard-coded into the application. This ensures that always the same paths are available in the user interface. Furthermore, custom paths with specific characteristics can be added and tested easily, which would be difficult to achieve with an actual pathfinding backend. We plan to conduct both qualitative studies, e.g., using the think-aloud method, as well as quantitative studies, e.g., using the system usability scale [17], with the testbed.

---

[5] https://svelte.dev (accessed 2023/09/05); the full tech stack is described in the provided Git repository[3].

[6] https://www.docker.com (accessed 2023/09/05)

[7] In a future end-to-end implementation, users will input a standard search query and the entities will be automatically extracted and linked, eliminating the need for manual entity selection.

[8] The provided Git repository[3] comprises thorough development instructions.
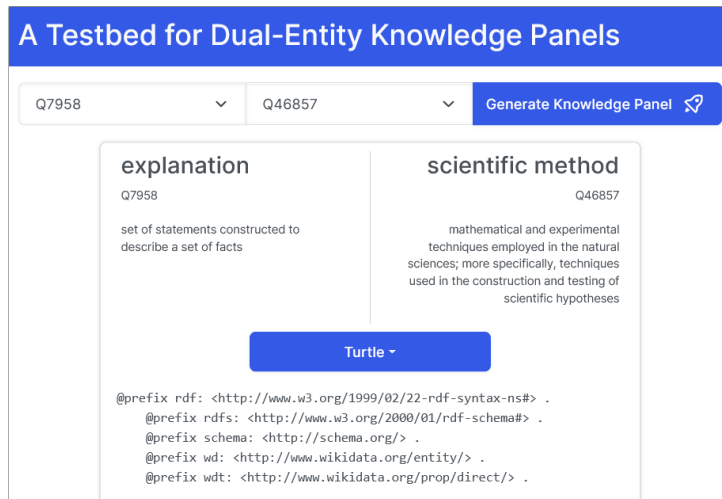
**Figure 2:** The user interface of the testbed after clicking the *Generate Knowledge Panel* button. In this screenshot, the lengthy Turtle document encoding the path is not shown completely.

## 4. Showcase of Presentation Formats

In addition to the *Turtle* presentation format, the four prototypical presentation formats depicted in Figure 3 have been implemented yet. *Arrow* corresponds to the presentation format used in Section 2 to give an example for a path in Wikidata. While this presentation format features the human-readable labels of entities and properties from Wikidata, it suffers from the prominent positioning of their proprietary IDs, which are supposedly not useful for most users. Next, *LLM* presents the path as a natural language explanation generated with ChatGPT. For this purpose, ChatGPT was asked to generate a description of the path encoded in the Turtle document. To mature this presentation format, the prompts provided to the LLM of choice need to be refined to optimize content, length, and structure of the description based on user feedback. That being said, the depicted explanation already possesses a decent quality. Both the third and the fourth presentation format use a graph-based visualization of the path but leverage different layouts. While *Graph: Circle* arranges the nodes in a circle, *Graph: Hierarchy* interprets the path as a hierarchy with respect to the direction of the properties. In both layouts, hovering over the path components further triggers a popup revealing the respective ID and description. Furthermore, the entities of the dual-entity query are highlighted in blue. Still, the latter format is supposedly easier to parse for users, especially when the properties express taxonomic relations. Facilitating the evaluation of such assumptions is a primary objective of the testbed.

Future work should not only evaluate and refine the showcased presentation formats but also investigate further options. Considering that KPs can comprise various media depending on the particular entity, even exotic presentation formats might be useful for some dual-entity queries. For example, AI-generated pictures could be leveraged to explain the relationship of certain entities. Wolfram|Alpha[9], an engine for factual query answering, also generates different visualizations comparing characteristics depending on the type of the queried entities. This

---

[9]https://www.wolframalpha.com (accessed 2023/09/05)

(a) The *Arrow* presentation format.



(b) The *LLM* presentation format.



(c) The *Graph: Circle* presentation format.



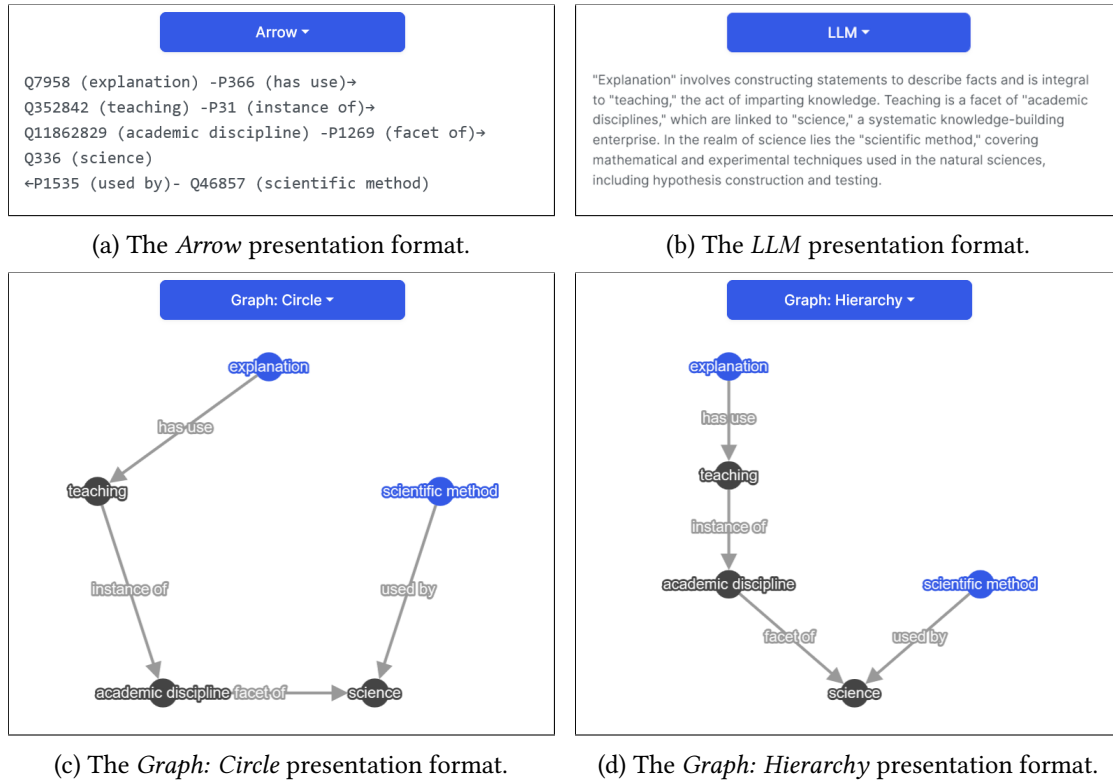(d) The *Graph: Hierarchy* presentation format.

**Figure 3:** The four currently implemented prototypical presentation formats.

includes maps with locations of geographic entities and tables comparing features of entities from a similar domain, among others. Context sensitivity is thus an important topic to be explored. Moreover, combinations of different presentation formats should also be considered.

## 5. Conclusion

The present paper introduced a testbed facilitating the implementation and evaluation of dual-entity KPs, i.e., KPs that are presented in response to dual-entity queries. In Section 4, a selection of prototypical presentation formats that convey an explanation of the relationship between the two entities of such queries was showcased. Even though this paper focused on paths from Wikidata as a basis for generating the explanations, the testbed can be adapted to operate on paths from other KGs if desired.

Complementary to the study of representation formats, another line of research is required that examines the usefulness of paths with respect to their characteristics. For example, paths with properties expressing taxonomic relationships might be more accessible to typical end users. In addition, paths exceeding a certain length might yield explanations with a lower perceived usefulness. The two lines of research must go hand in hand since path characteristics influence the applicability of presentation formats, as well. For instance, graph-based presentation formats are not suitable for illustrating paths beyond a certain length.

# References

[1] A. Singhal, Introducing the Knowledge Graph: Things, not Strings, Official Search Blog (2012). https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html (accessed 2023/09/05).

[2] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. F. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Comput. Surv. 54 (2022) 71:1–71:37. URL: https://doi.org/10.1145/3447772.

[3] L. Martin, BiPaSs: Further investigation of fast pathfinding in Wikidata, in: SEMANTiCS 2023: 19th International Conference on Semantic Systems, Leipzig, Germany, September 20-22, 2023, Proceedings, (accepted) 2023.

[4] L. Martin, J. H. Boockmann, A. Henrich, Fast pathfinding in knowledge graphs using word embeddings, in: U. Schmid, F. Klügl, D. Wolter (Eds.), KI 2020: Advances in Artificial Intelligence - 43rd German Conference on AI, Bamberg, Germany, September 21-25, 2020, Proceedings, volume 12325 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 305–312. URL: https://doi.org/10.1007/978-3-030-58285-2_27.

[5] D. Vrandecic, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Commun. ACM 57 (2014) 78–85. URL: https://doi.org/10.1145/2629489.

[6] R. Reinanda, E. Meij, M. de Rijke, Knowledge graphs: An information retrieval perspective, Found. Trends Inf. Retr. 14 (2020) 289–444. URL: https://doi.org/10.1561/1500000063.

[7] R. Cyganiak, D. Hyland-Wood, M. Lanthaler, RDF 1.1 concepts and abstract syntax, W3C Recommendation (2014). https://www.w3.org/TR/rdf11-concepts (accessed 2023/09/05).

[8] G. Kasneci, S. Elbassuoni, G. Weikum, MING: mining informative entity relationship subgraphs, in: D. W. Cheung, I. Song, W. W. Chu, X. Hu, J. Lin (Eds.), Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009, ACM, 2009, pp. 1653–1656. URL: https://doi.org/10.1145/1645953.1646196.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, 3rd Edition, MIT Press, 2009. URL: http://mitpress.mit.edu/books/introduction-algorithms.

[10] D. Beckett, T. Berners-Lee, E. Prud'hommeaux, G. Carothers, RDF 1.1 turtle - terse rdf triple language, W3C Recommendation (2014). https://www.w3.org/TR/turtle (accessed 2023/09/05).

[11] I. Herman, G. Melançon, M. S. Marshall, Graph visualization and navigation in information visualization: A survey, IEEE Trans. Vis. Comput. Graph. 6 (2000) 24–43. URL: https://doi.org/10.1109/2945.841119.

[12] F. Beck, M. Burch, S. Diehl, D. Weiskopf, A taxonomy and survey of dynamic graph visualization, Comput. Graph. Forum 36 (2017) 133–159. URL: https://doi.org/10.1111/cgf.12791.

[13] J. Nielsen, Enhancing the explanatory power of usability heuristics, in: Conference on Human Factors in Computing Systems, CHI 1994, Boston, Massachusetts, USA, April 24-28, 1994, Proceedings, 1994, pp. 152–158. URL: https://doi.org/10.1145/191666.191729.

[14] S. R. Midway, Principles of effective data visualization, Patterns 1 (2020) 100141. URL: https://doi.org/10.1016/j.patter.2020.100141.

[15] A. Gatt, E. Krahmer, Survey of the state of the art in natural language generation: Core tasks, applications and evaluation, J. Artif. Intell. Res. 61 (2018) 65–170. URL: https://doi.org/10.1613/jair.5477.

[16] OpenAI, GPT-4 technical report, CoRR abs/2303.08774 (2023). URL: https://doi.org/10.48550/arXiv.2303.08774.

[17] J. Brooke, SUS: a "quick and dirty" usability scale, Usability evaluation in industry 189 (1996) 189–194.