OnSET: Ontology and Semantic Exploration Toolkit

Benedikt Kantz
benedikt.kantz@tugraz.at
University of Technology Graz
Graz, Austria

Kevin Innerebner
innerebner@tugraz.at
University of Technology Graz
Graz, Austria

Peter Waldert
peter.waldert@tugraz.at
University of Technology Graz
Graz, Austria

Stefan Lengauer
s.lengauer@tugraz.at
University of Technology Graz
Graz, Austria

Elisabeth Lex
elisabeth.lex@tugraz.at
University of Technology Graz
Graz, Austria

Tobias Schreck
tobias.schreck@tugraz.at
University of Technology Graz
Graz, Austria

Abstract

Retrieval over knowledge graphs is typically performed using specialized, complex query languages such as SPARQL. We propose a novel system, Ontology and Semantic Exploration Toolkit (OnSET), that allows novice users to quickly build queries with visual user guidance provided by topic modeling and semantic search throughout the application. OnSET enables users without prior knowledge of the ontology or networked knowledge to start exploring topics of interest over knowledge graphs, including the retrieval and detailed exploration of prototypical sub-graphs and their instances. Existing systems either focus on direct graph exploration or do not foster further exploration of the result set. We, however, provide a node-based editor that can extend these missing properties of existing systems to support search over large ontologies with sub-graph instances. Furthermore, OnSET combines efficient and open platforms to deploy the system on commodity hardware.

CCS Concepts

• **Information systems → Users and interactive retrieval**; *Information extraction*; *Ontologies.*

Keywords

Ontology, visualization, graph retrieval, natural language, user guidance

ACM Reference Format:
Benedikt Kantz, Kevin Innerebner, Peter Waldert, Stefan Lengauer, Elisabeth Lex, and Tobias Schreck. 2025. OnSET: Ontology and Semantic Exploration Toolkit. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3726302.3730148

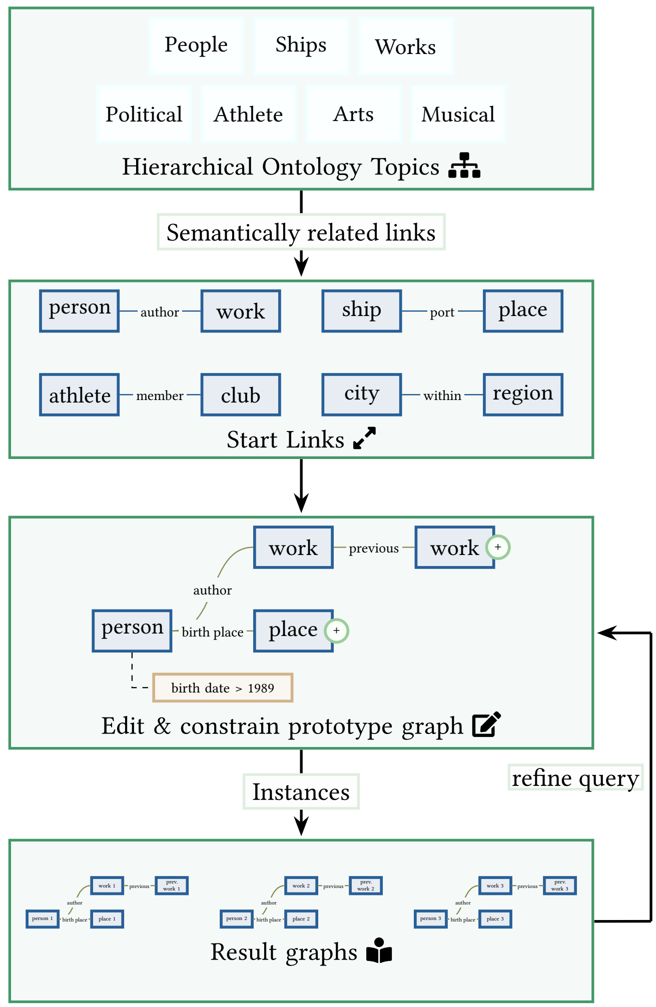

Figure 1: OnSET user flow. The user can select topics of interest and retrieve possible start links. These links are then expanded & constrained within an editor, which finally retrieves different instances of the searched graph. A demo video is accessible at https://cloud.tugraz.at/index.php/s/djdayXSSWAX4ajt, and the source code is available at https://github.com/Dakantz/OnSET

1 Introduction

Information retrieval from knowledge graphs is typically performed using specialized languages, such as SPARQL [9, 17].

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3730148

These query languages, however, require specialized knowledge of both the syntactical nature of the query language and the schema of the knowledge graph. This additional barrier of entry can be challenging for non-experts of these languages and first-time users, hindering easy exploration of these rich knowledge bases, such as DBpedia [11].

To further the use of ontologies and knowledge graphs by non-expert users, existing systems already provide natural language or visual interfaces, making these applications more accessible to them. We relate these works to ours in Section 2. Compared to existing systems, our system enhances the initial exploration phase of ontologies.

Our system, OnSET, combines the existing interface paradigms to foster approachable exploration of ontologies. Our system incorporates concepts from user guidance [16] to facilitate the exploration of large ontologies while maintaining a user-friendly approach. None of the related works consider the missing knowledge of users who may need guidance towards a first information need. We solve this initial guidance problem using topic modeling over the ontology, an approach originating from Natural Language Processing (NLP). We furthermore extend the expansion of our prototype graph[1] with a semantic search interface that reduces the barrier of entry, users do not need prior knowledge of the ontology, only semantic information derived from the topics presented initially. Furthermore, we expand the notion of dynamic results by providing instances of our prototype graph as small multiples, which can be inspected with further detail on demand. We utilize advances in NLP as the basis for the search and exploration capabilities within OnSET. Our application builds on open systems, starting with our SPARQL databases system qlever [1], over the topic modeling approach based on BERTopic [8] and finally using open and efficient Language Models (LMs) [10, 18].

2 Related Work

Existing retrieval systems already aim to reduce these barriers to entry. Some of these include natural language querying approaches [5, 12, 13], others build towards visual interfaces for building queries over knowledge bases [3, 6, 7, 14, 19].

Natural language interfaces focus on the ease of use and adaption of users' information needs as text, or, to some extent, incorporate NLP concepts [12, 13]. The *NLP Engine* [12] aims to convert any natural language query to a SPARQL query for further processing using multiple translational layers. The system focuses on query mappings and assumes some prior knowledge of the schema users within the data to formulate queries in natural language. *Graph Query Suggestion* [13] expands the initial query graph of a user by suggesting related links using traditional Information Retrieval (IR) approaches. Furthermore, they build on the notion of example graphs, where the user builds a graph that serves as an example or prototype to retrieve similar structures from the knowledge graph.

On the other hand, the visual querying approaches focus on providing approachable systems to facilitate the use of knowledge graphs for non-experts. *RDF Explorer* [19] approaches this task by providing a graph editor, enabling the creation of graph examples to retrieve matching instances from the knowledge graph. The system provides users with query expansion options throughout the application, displaying dynamic results as queries are built. The authors relate their work to previous query builders, notably *Smeagol* [3]. This alternative follows similar exploration and retrieval paradigms but does not offer a comparable number of SPARQL features. *KGVQL* [14] defines a novel visual query language to ease the transformation between the visual querying and result set, at the cost of disregarding explorative approaches, favoring the proposed transformation approach to convert between data examples and queries. *Rhizomer* [7] approaches the exploration of knowledge bases by providing different user interfaces to explore only at the top-level, graph-level, or only at the instance level of a single type. *Sparnatural* [6] simplifies many of these exploratory approaches into a tree-based approach that directly yields tabular results, thereby limiting easier result set exploration. Similarly, *SPARKLIS* [5] approaches the topic by representing the query as a single natural language query expressed as a template over the ontology and combines the creation of the query with a small user interface system. Neither of these systems offers fuzzy search interfaces or initial user guidance to support novice users in their explorative search.

3 Methodology

OnSET builds on task-driven user guidance design principles [16]. To this end, we first define the *target* of the retrieval task as a subgraph of the ontology. This graph should be retrieved exploratively, i.e., the user can search for possible instances given some query. We aim to guide the user by *leading* them towards subsets of interest and responding to user cues as they explore the ontologies.

We incorporate user guidance into the initial exploration step using BERTopic [8] to group the ontology into hierarchical topics and present the user with an overview of possible ontology aspects. Topic modeling, typically performed on text documents, involves representing each class within the ontology as a single document, utilizing templates for the class, its parent classes, and the properties associated with the class. The resulting topics are then labeled using a LM to provide concise textual labels for each topic within the ontology. The novice user can, therefore, choose one or more topics of interest, which are then used to query for start links and classes. This retrieval of start elements uses the averaged semantic embedding of the selected topics and semantic embeddings of textual representations of all links and classes within the ontology.

Once the user has chosen a start link based on the suggested set from the topic selection, the graph-building process is started. Our prototype graph-building system, shown in Figure 3, is inspired by notions from prior work [6, 19] but extends them in crucial ways. Our significant contribution is the inclusion of semantic search for all outgoing links between classes, enabling fuzzy search over the ontology. This guidance approach is more flexible than existing link expansion approaches [13], as it can respond to users' information needs in more flexible ways and is not bound to prior selections, giving the user more freedom while still adhering to the ontology. This semantic search is also applied to the attribute constraints, which can be searched similarly and applied to each entity. The notion of the prototype graph is closely related to the concept of

[1]The term *example query* is used by related works [13, 19] to describe their approach of providing an instance from the knowledge graph. In contrast, our query paradigm builds on the *prototype* for an instance.

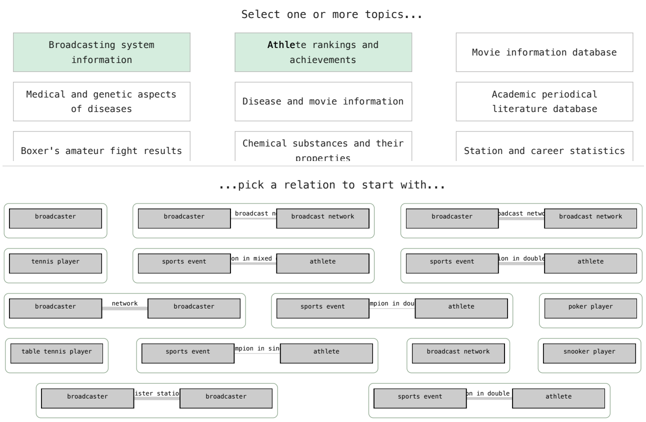

Figure 2: The initial selection process of OnSET presents the user with a set of topics. The user can select one or more topics to initiate the exploration, choose a starting link, and then edit the graph in the subsequent steps.

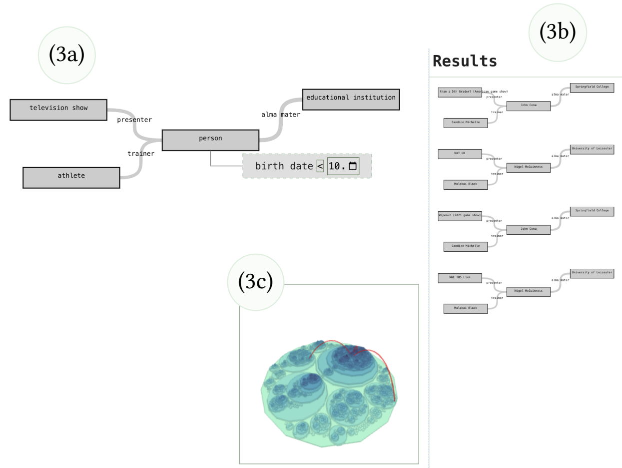

Figure 3: The query can be built using a straightforward interactive process. The user can add any allowed link within the ontology, with a visual indication of its prevalence within the knowledge graph shown by the link width. Users can also add constraints to the nodes. (3a) The tool immediately provides visual results (3b) and provides a visual indicator of the explored classes and links using a small three-dimensional circle packing "minimap" in the bottom right (3c).

Basic Graph Patterns (BGPs), with the addition of more constraints to the schema provided by the ontology and constraints placed upon properties by the user.

The resulting prototype graph (3a) is then utilized in three ways, similar to the levels of García et al. [7], but integrated into the core retrieval process. First, the graph is used on top of a 3D circle-packing visualization of the ontology to illustrate how the classes are distributed over the ontology with respect to its class hierarchy (3c). Second, the prototype graph is used to generate a SPARQL query to retrieve the intended instance set. This instance set populates the third use case of the graph, where we show small instances of the initial prototype graph. (3b). These instances, or result sets, can then be inspected, and the properties of the retrieved instances can be explored individually. The presentation of smaller, visually similar instances also differentiates our system from existing approaches, as we visually relate the result set and prototype graph.

4 Implementation

The outlined concepts are integrated into our system, OnSET, focusing on fast user responses even on larger ontologies. We furthermore base our entire stack on open-source systems, allowing institutions or users to start and tweak their systems.

To achieve our first goal of fast user responses, we only compute the topic modeling and embeddings of links and classes on the first startup with a specific ontology. We store our resulting hierarchical topic map and embeddings in PostgreSQL[2] with the help of the pgvector extension [3], which enables fast retrieval given a query embedding. To generate these embeddings quickly, even on commodity hardware, we use the `stella_en_400M_v5`[4] model, which is, at the time of submission, the best-performing smaller model w.r.t. the massive text embedding benchmark [15]. The precomputed topics are generated using the same embedding model, and additional topic labels are generated using the Hermes Llama 3.2 8B model [18].

While these efforts improve the query time for building the prototype graph, live updates of the result set require a similarly fast system. To achieve those fast responses, even on more complex queries and on commodity hardware, we use qlever [1], a SPARQL query engine that outperforms most existing engines in both speed and system requirements. This speedup enables our system to serve and display updates as the user builds their query, aiding the user in retrieving non-empty sets and showing intermediate results to guide the search even further.

Our user interface builds on Vue.js[5], in combination with three.js[6] and D3.js [2]. All the used database systems and models are open-source and open-weight, providing state-of-the-art performance in their respective fields while still being able to run on commodity hardware.

5 Case study

We present OnSET in the scope of two case studies for explorative search over two different ontologies. First, we show how a novice user might start their exploration of DBpedia [11] to discover interesting facts and relations. Our second use case covers the Brainteaser Ontology (BTO) [4] and how experts within a field might approach more specialized ontologies.

5.1 Exploring DBpedia

DBpedia [11] is a curated ontology and knowledge graph derived from Wikipedia, providing general linked information. Querying

[2]https://www.postgresql.org/
[3]https://github.com/pgvector/pgvector
[4]https://huggingface.co/NovaSearch/stella_en_400M_v5
[5]https://vuejs.org/
[6]https://threejs.org/

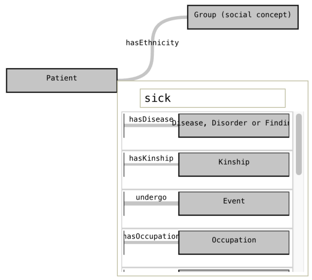

Figure 4: The search interface to retrieve semantically similar links, on the example of BTO [4].

and exploring this knowledge base typically requires the use of the SPARQL language. Our visual exploration toolkit, OnSET, enables the novice user to start exploring the knowledge base immediately through the presented topics.

An example of such an exploration flow could be the initial selection of the topic "Broadcasting system information" and "Athlete rankings and achievements", as seen in Figure 2. This selection queries our system for links similar to the selected topic and displays start link suggestions to the user. The user can, therefore, start exploring DBpedia without prior knowledge of the classes and links contained in it, hinting the user at possible links within the ontology. Next, the user selects a link from the suggested list, initiating the process of building the prototype graph. The user then adds links and constraints to the prototype graph, drilling down towards a specific result set of interest—in this case, persons who trained an athlete and presented a television show, and where they were educated. OnSET guides the user towards non-empty sets by indicating prevalence in the knowledge graph through link strength. The user can assume that the result set is not empty due to the strong links between all classes on the prototype graph, which can be immediately verified as the result set is updated directly. While building the graph, the "minimap" in the corner of Figure 3 is dynamically updated to indicate which regions within the class hierarchy are covered. In this case, two links span the class hierarchy while one link, the athlete's trainer, covers only the local hierarchy within the person region.

5.2 Exploring BTO

Our second example outlines the exploratory search over the BTO [4], an ontology and knowledge graph that contains semantic knowledge about patients, caregivers, and their diagnoses. The explorative process for an interested party starts similarly to the use case above, but is presented with different topics.

The user might be interested in diseases associated with certain relationsships, so they start their search with the topic "Kinship types and relations", starting with the link "hasEthnicity". This single relation is too general, so they search for sicknesses, as seen in Figure 4. Using the semantic search capabilities, they find the relation "hasDisease" within the ontology and add them to their prototype graph. Nevertheless, this relatively small prototype graph allows the user to compare and search over these links of interest in the result set.

6 Future Work

OnSET provides a low barrier of entry for novice users to knowledge graph exploration. We intend to expand the breadth of queries that users can express through our system in the future. A current limitation of our system is the inability to specify complete graphs, as the user can, for the time being, only build tree-like graphs, while closed graphs might be of interest for more advanced or intricate use cases.

We intend to refine the constraint application process as the filtering strength of the properties of an instance is not yet clearly visually defined. This refinement could aid the user in exploring and retrieving data more attuned to their need and assist in non-empty result retrieval. Another interesting avenue is the extension towards optional parts of the prototype graph, both in the form of links and constraints, to facilitate more fuzzy retrieval. Another missing aspect of the constraint-building process is the consideration of missing properties, where a search over multimodal data types, such as spatial or image data, could be interesting.

Other visual cues to ease the query-building process could be of interest, such as result set changes upon adding or removing query parameters as a what-if visual cue in all affected results and "minimap".

7 Conclusion

The presented IR system, OnSET, allows novice users without any prior information about the knowledge base to build queries and explore it in an integrated manner. We first present related systems that share a similar aim of enabling non-expert users to construct SPARQL queries and explore knowledge graphs. Our approach, however, differs in that we utilize initial guidance approaches to lower the barrier of entry even further by providing semantic search for both the initial link search and the prototype graph expansion, and ultimately, immediate result feedback is built right into the interface. We also provide an overview of the ontology to help relate the current query to the entire ontology.

We emphasize the use of specialized open database systems to provide fast response times, enabling interactive and immediate result exploration, even for minor changes to the prototype graph. We finally demonstrate two possible explorative user flows using OnSET to inspire further use cases.

Acknowledgments

This work is partially supported by the HEREDITARY Project, as part of the European Union's Horizon Europe research and innovation programme under grant agreement No GA 101137074, the Austrian Science Fund (FWF) 10.55776/COE12, Cluster of Excellence Bilateral Artificial Intelligence and the FFG HybridAir project #FO999902654.

References

[1] Hannah Bast and Björn Buchhold. 2017. QLever: A Query Engine for Efficient SPARQL+Text Search. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (Singapore, Singapore) *(CIKM '17)*. Association for Computing Machinery, New York, NY, USA, 647–656. doi:10.1145/3132847.3132921

[2] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D3: Data-Driven Documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)* (2011). http://vis.stanford.edu/papers/d3

[3] Aaron Clemmer and Stephen Davies. 2011. Smeagol: A "Specific-to-General" Semantic Web Query Interface Paradigm for Novices. In *Database and Expert Systems Applications*, Abdelkader Hameurlain, Stephen W. Liddle, Klaus-Dieter Schewe, and Xiaofang Zhou (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 288–302.

[4] Guglielmo Faggioli, Laura Menotti, Stefano Marchesin, Adriano Chió, Arianna Dagliati, Mamede de Carvalho, Marta Gromicho, Umberto Manera, Eleonora Tavazzi, Giorgio Maria Di Nunzio, Gianmaria Silvello, and Nicola Ferro. 2024. An extensible and unifying approach to retrospective clinical data modeling: the BrainTeaser Ontology. *Journal of Biomedical Semantics* 15, 1 (Aug. 2024). doi:10.1186/s13326-024-00317-y

[5] Sébastien Ferré. 2017. SPARKLIS: An Expressive Query Builder for SPARQL Endpoints with Guidance in Natural Language. *Open Journal Of Semantic Web* 0 (2017). https://inria.hal.science/hal-01485093

[6] Thomas Francart. 2023. Sparnatural: a visual knowledge graph exploration tool. In *European Semantic Web Conference*. Springer, 11–15.

[7] Roberto García, Juan-Miguel López-Gil, and Rosa Gil. 2022. Rhizomer: Interactive semantic knowledge graphs exploration. *SoftwareX* 20 (2022), 101235. doi:10.1016/j.softx.2022.101235

[8] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794* (2022).

[9] Thomas R. Gruber. 1995. Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies* 43, 5 (1995), 907–928. doi:10.1006/ijhc.1995.1081

[10] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2024. Matryoshka Representation Learning. arXiv:2205.13147 [cs.LG] https://arxiv.org/abs/2205.13147

[11] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, S. Auer, and Christian Bizer. 2015. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6 (2015), 167–195. https://api.semanticscholar.org/CorpusID:1181640

[12] Chuan Lei, Fatma Özcan, Abdul Quamar, Ashish R Mittal, Jaydeep Sen, Diptikalyan Saha, and Karthik Sankaranarayanan. 2018. Ontology-Based Natural Language Query Interfaces for Data Exploration. *IEEE Data Eng. Bull.* 41, 3 (2018), 52–63.

[13] Matteo Lissandrini, Davide Mottin, Themis Palpanas, and Yannis Velegrakis. 2020. Graph-Query Suggestions for Knowledge Graph Exploration. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 2549–2555. doi:10.1145/3366423.3380005

[14] Pengkai Liu, Xin Wang, Qiang Fu, Yajun Yang, Yuan-Fang Li, and Qingpeng Zhang. 2022. KGVQL: A knowledge graph visual query language with bidirectional transformations. *Knowledge-Based Systems* 250 (2022), 108870. doi:10.1016/j.knosys.2022.108870

[15] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. MTEB: Massive Text Embedding Benchmark. *arXiv preprint arXiv:2210.07316* (2022). doi:10.48550/ARXIV.2210.07316

[16] Ignacio Pérez-Messina, Davide Ceneda, and Silvia Miksch. 2023. A Methodology for Task-Driven Guidance Design. In *EuroVis Workshop on Visual Analytics (EuroVA)*, Marco Angelini and Mennatallah El-Assady (Eds.). The Eurographics Association. doi:10.2312/eurova.20231094

[17] Andy Seaborne, Ruben Taelman, Gregory Williams, Olaf Hartig, and Thomas Pellissier Tanon. 2024. *SPARQL 1.2 Query Language*. W3C Working Draft. W3C. https://www.w3.org/TR/2024/WD-sparql12-query-20241227/.

[18] Ryan Teknium, Jeffrey Quesnelle, and Chen Guang. 2024. Hermes 3 Technical Report. arXiv:2408.11857 [cs.CL] https://arxiv.org/abs/2408.11857

[19] Hernán Vargas, Carlos Buil-Aranda, Aidan Hogan, and Claudia López. 2019. RDF explorer: A visual SPARQL query builder. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part I 18*. Springer, 647–663.