State-of-the-Art in Knowledge Graph Exploration

Rana İşlek[1] and Berat Furkan Kocak[1]
[1]*UPC Universitat Politècnica de Catalunya*

Knowledge Graphs (KGs) are structured representations of information, enabling intelligent applications such as data analytics and recommender systems. This report presents a comprehensive review of state-of-the-art methods in Knowledge Graph Exploration, addressing key tasks, challenges, and recent advancements. We explore various approaches including summarization, profiling, exploratory data analytics, exploratory search and recommendation systems, drawing from the latest literature. By identifying common themes and contrasting techniques, this report highlights solutions to improve the accessibility and effectiveness of KG exploration for both expert and non-expert users.

I. INTRODUCTION

The use of semantic web technologies has resulted in the creation of knowledge graphs depicted as triples, enabling exploration across various domains. These graphs facilitate semantic search, the integration of data, and thorough data analysis. Although semantic links between entities can help users explore content, they can also result in a large number of overwhelming choices that may confuse or frustrate users, especially those unfamiliar with the subject[1].

Knowledge Graph Exploration is a vast research area. Hence we focus on the research question ”How can we improve the accessibility and effectiveness of Knowledge Graph exploration for both expert and non-expert users?”. By consolidating findings from various studies, this report aims to provide a clear understanding of effective KG exploration techniques and their applications, ultimately enhancing the usability of KGs for diverse user groups.

This report reviews the current state-of-the-art in KG exploration, focusing on and giving explanation of their key methods. We examine challenges they identify in the literature, including complexity, heterogeneity, and scalability of KGs, and discuss their novel proposals designed to overcome these obstacles. We identified the most important six different methodology addressing our research question that presented in the methods section.

II. LITERATURE REVIEW

In the paper ”Knowledge Graph Exploration Systems: Are We Lost?”, the authors highlight that current Knowledge Graph (KG) exploration systems are inadequate for modern use cases, despite widespread adoption by companies like Amazon, Google, and Microsoft. These systems struggle with understanding KG structures, identifying relevant portions, and extracting insights due to the complexity and heterogeneity of KGs. Authors focuse on key challenges including heterogeneity, evolution, vagueness, and scale. The goal of the paper is to identify requirements for effective KG exploration systems and highlight promising research directions.[2]

”KGScope: Interactive Visual Exploration of Knowledge Graphs with Embedding-based Guidance” defines current KG exploration systems as inadequate due to the complexity and size of KGs. It discusses that existing systems typically offer query boxes or search interfaces assuming users have specific objectives and understanding of the data and how these systems lack graph-centric analyses and do not leverage rich semantic information within KGs. The authors address the ineffectiveness of current systems in supporting users, particularly data analysts, in uncovering and summarizing insights from KGs. As a solution, KGScope is proposed to support interactive visual explorations with embedding-based guidance, helping users derive insights with intuitive visualizations and tools. [4]
”Search Box System to Improve User Interaction in Knowledge Graph Searches: A Solution for Users Without Expert Skills” focuses on the complex nature of KG exploration in Wikidata, particularly for inexperienced users. The authors point out that users often navigate through multiple entities and perform numerous clicks and searches to find property names and relevant information, which is not user-friendly. The paper addresses the difficulty non-expert users face in exploring and entering new data into Wikidata. The paper proposes a new search box system for Wikidata that allows non-expert users to perform exploratory searches based on chained entities without navigating through multiple pages.[5]

In "A Survey on Extractive Knowledge Graph Summarization: Applications, Approaches, Evaluation, and Future Directions", the authors define KG summarization as an emerging and crucial task due to the continuous growth of large KGs. The paper highlights the importance of creating compact, information-dense subgraphs (summaries) to facilitate tasks like KG profiling, query optimization, search, and exploration. The main problem addressed is the challenge of discovering and selecting suitable KGs for reuse due to their large size and topic diversity. Authors' goal is to provide a comprehensive survey of extractive KG summarization, covering its applications, methods, evaluation metrics, and future research directions.[3]

The Paper "Exploring Importance Measures for Summarizing RDF/S KBs", tackles the challenge of efficiently extracting high-quality summary graphs from RDF/S Knowledge Bases (KBs), aiming to identify the most crucial nodes while minimizing false positives. The authors reevaluate six well-established graph theory measures, adapting them specifically for RDF/S KBs to solve this issue. They also approach the task of connecting significant nodes as a Graph Steiner Tree Problem (GSTP), utilizing approximations and heuristics to improve algorithm execution speed.

The main problem addressed is the difficulty users encounter when formulating queries, as they need to thoroughly examine the entire schema to find relevant elements and data. To mitigate this, the paper focuses on developing methods and tools that enable rapid understanding and exploration of data sources via automatic ontology summarization. The objective is to create efficient methods for producing high-quality summaries that accurately represent a sub-schema graph with the most important nodes, making data exploration and query formulation easier. [6]

"AttnIO: Knowledge Graph Exploration with In-and-Out Attention Flow for Knowledge-Grounded Dialogue" focuses on the correct knowledge selection problem in conversational context in dialogue systems. They state that previous works utilized path traversal on external knowledge graphs to solve this problem;however, they state an improvement in performance by introducing two directions on attention flows. They also state that the AttnIO is capable of generating proper knowledge path when paths are not available and only the destination nodes are given as labels. [7]

"Example-Driven Exploratory Analytics over Knowledge Graphs" focuses on another niche in knowledge graph exploration by enabling OLAP queries over KG with reverse-engineering. They state that current methods allow entity explorations either without supporting analytical queries or the ones focusing on OLAP queries do so by converting the graph to a relational model which is highly inefficient. This paper introduces Re2xOLAP, the first comprehensive interactive method for reverse-engineering and refining RDF exploratory OLAP queries over knowledge graphs that contain statistical data. Re2xOLAP supports analytical queries and leveraging the RDF Data Cube Vocabulary for describing multi-dimensional data in knowledge graphs. This approach enables users to perform exploratory analytics on knowledge graphs without writing queries. This way, they also enable non-experts in this area to easily perform OLAP queries on knowledge graphs. [8]

"KPRLN: deep knowledge preference-aware reinforcement learning network for recommendation" focuses on user preference based recommender systems. We included this paper in our state-of-art because they provide a very novel approach to find user specific subgraphs out of a knowledge graph, and utilize reinforcement learning to enhance recommendation accuracy by generating a weighted knowledge graph. KPRLN constructs paths connecting users' historical interactions within the knowledge graph, enabling the learning of user-specific preference characteristics for each entity–relation pair. [9]

III. METHODS

Knowledge graph exploration has advanced significantly, employing various state-of-the-art approaches to enhance understanding, uncover relationships, and facilitate decision-making processes across diverse domains. In the paper "Knowledge Graph Exploration Systems: are we lost?": authors give a nice introduction to this topic and discuss the challenge of knowledge graph exploration by providing a spectrum of features showing state-of-the-art solutions including various methods from different papers to handle this challenge.[2]

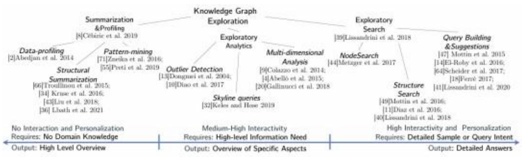

FIG. 1: Taxonomy of KG Exploration techniques and their positioning on the spectrum of features. [2]

As one can see in Figure 1, authors classified current solutions as "summarization & profiling", "exploratory analytics", and "query building & suggestions", and three different subcategories for each main category. Furthermore, the authors of this paper identify four main challenges that graph management systems face while handling KG exploration: heterogeneity, evolution, vagueness, and scale; while pointing out that heterogeneity is the most significant one. In Figure 2, one can find a nice representation showing the connections between tasks, goals, challenges, and operations that characterize KG exploration systems according to this paper.

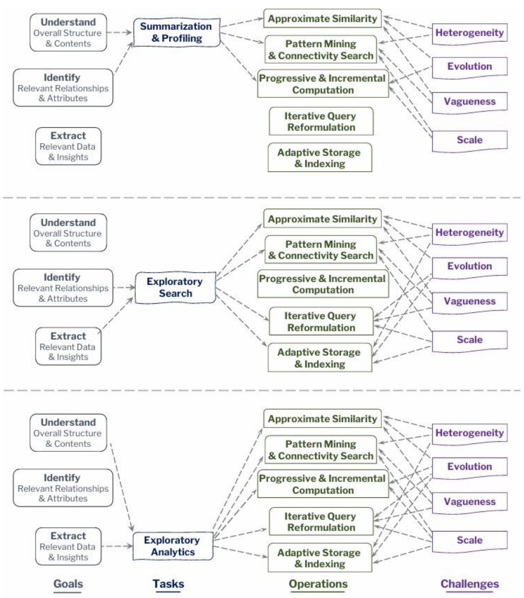

FIG. 2: The connections between tasks, Goals, Challenges, and Operations characterizing KG exploration systems.

[2]

Following section discusses the methods and techniques identified in the reviewed papers for improving KG exploration, including their use cases and results.

A. Approximate Similarity Search

Approximate similarity search techniques allow users to search for nodes or edges similar to a given example, considering domain-specific similarity measures. This method requires specialized algorithms to manage the high computational costs associated with these operations.

In the current state-of-art many importance measures from graph theory exist in order to detect the most important nodes residing in a knowledge graph. The paper *Exploring Importance Measures for Summarizing RDF/S KBs* compares the accuracy of most known six of them, namely Degree , the Betweeness, the Bridging Centrality, the Harmonic Centrality, the Radiality and the Ego Centrality measures adapting them for RDF/S KBs to consider instance information as well.

Table I shows the complexity of each measure.

Measure	Complexity
Degree (DE)	$O(VS + ES)$
Betweeness (BE)	$O(VS \cdot (VS + ES))$
Bridging Centrality (BC)	$O(VS \cdot (VS + ES))$
Harmonic Centrality (HC)	$O(VS \cdot (VS + ES))$
Radiality (RA)	$O(VS \cdot (VS + ES))$
Ego Centrality (EC)	$O(VS + ES)$

TABLE I: Complexity of Centrality Measures

In order to compare the performances, they normalize the results to a scale from 0 to 1. This gives an AIM(adapted importance measure), helps selecting top-k important nodes of a directed schema graph (also known as terminals in graph theory).

After finding nodes, creating connections between them is also a problem. The latest approaches try identifying maximum cost spanning tree (MST) in the graph and then link the most important nodes accordingly [11]. However, they state that main problem with this approach is although the MST identifies the paths with the maximum weight in the whole graph, the paths selected out of the MST might not maximize the weight of the selected summary. Hence, they model this as a graph steiner-tree problem, and use approximation SDIST,CHINS and HEUM to optimize either the insertion of single component or connection of components using shortest paths. Their comprative worst-case complexities for linking most important nodes are given in Table II.

Algorithm	Weighted graph	Un-weighted graph
MST	$O(E \cdot \log V)$	$O(\|V+E\|)$
SDISTG	$O(Q \cdot \|V \log V\|)$	$O(Q \cdot \|V+E\|)$
CHINS	$O(Q \cdot \|V \log V\|)$	$O(Q \cdot \|V+E\|)$
HEUM	$O(V \cdot \|V \log V\|)$	$O(V \cdot \|V+E\|)$

TABLE II: Time complexities of algorithms on graphs

Figure 3 shows the comparative results of the experiment for different algorithms and measures.

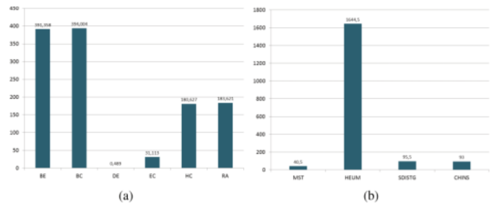

FIG. 3: The average execution times of the importance measures (msec) (a) and the algorithms for linking those nodes (msec) (b)
[6]

In conclusion, they state that the adaptation of Betweeness is the winner in all cases. In addition, they show that the Steiner-Tree approximation algorithms introduce less additional nodes to the result schema graph. Moreover, CHINS seems to be the best choice in terms of the quality of the generated [6]

B. Pattern Mining and Connectivity Search

Pattern mining and connectivity search methods identify relevant structures and patterns within the knowledge graph, often dealing with underspecified queries.

One of the novel solutions utilizing this method is provided by *KPRLN: deep knowledge preference-aware reinforcement learning network for recommendation*. They utilize the user's historical interaction items in the knowledge graph to create attention to specific parts of the KG, enhancing the recommendation accuracy provided to the user.

As each user have different preferences when choosing different items for KG exploration as well as understanding the preferences is not an easy task. Therefore they state that it is not enough to simply calculate user preference for the type of relationship but learn the preference features based on the user-entity-relations. User preferences can be personalized as weights on KG.

Their learning process is as follows: The deep reinforcement learning approach navigates the knowledge graph through cluster expansion and devises feedback rewards using hierarchical propagation routes. Simultaneously, the deep reinforcement learning agent adjusts the edge weights globally across the knowledge graph based on the anticipated returns for each connection. Within the recommendation prediction component, an attention mechanism is implemented to diffuse users' higher-order interests throughout the knowledge graph and blend user and item features for predictive purposes. The RL model establishes paths between clusters, and upon identifying a cluster-to-cluster path, it retroactively updates the associated nodes within the originating cluster to connect them with all nodes in the destination cluster [9]. Figure 4 showcases the overall framework of KPRLN.

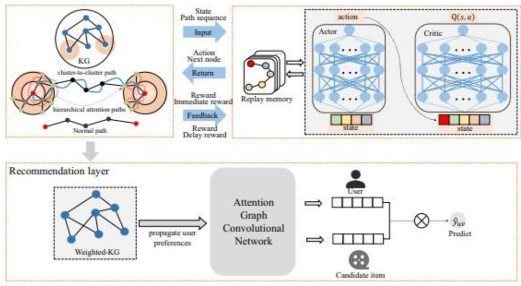

FIG. 4: Illustration of the KPRLN framework
[9]

The paper presents the results of CTR prediction and top-K recommendation of KPRLN and other baselines. Experiment is performed on Movielens-1M and Last.FM datasets which contain movie and music information alongside with user reviews. The experiment is conducted with the hyperparameter settings shown in Fig III and the comparative results with other novel methods are provided in IV

Dataset	d	η	N	H	Batch size
Movielens-1M	64	1×10^{-3}	421	10^{-5}	1024
Last.FM	16	5×10^{-4}	811	10^{-4}	256

TABLE III: Hyper-parameter settings for different datasets
[9]

Dataset	Methods	ACC	Impr	F1	Impr
Movielens-1M	LibFM	0.812	4.5%	0.819	4.0%
	PER	0.664	19.3%	0.673	18.6%
	CKE	0.742	11.5%	0.742	11.7%
	Ripplenet	0.844	1.3%	0.848	1.1%
	KGCN	0.840	1.7%	0.843	1.6%
	HAGERec	0.847	1.0%	0.847	1.2%
	KPRLN	**0.857**	–	**0.859**	–
Last.FM	LibFM	0.711	4.2%	0.710	3.7%
	PER	0.596	15.7%	0.596	15.1%
	CKE	0.673	8.0%	0.673	7.4%
	Ripplenet	0.691	6.2%	0.702	4.5%
	KGCN	0.731	2.2%	0.721	2.6%
	HAGERec	0.743	1.0%	0.734	1.3%
	KPRLN	**0.753**	–	**0.747**	–

TABLE IV: Performance of different methods on Movielens-1M and Last.FM datasets
[9]

C. Theme-Based and Keyword Search

Theme-based and keyword search approaches improve knowledge graph exploration through intuitive methods like theme identification and keyword searches. In this context, *Search Box System to Improve User Interaction in Knowledge Graph Searches* focuses on live interaction and keyword suggestions to simplify user searches.[5]

The authors of this paper propose a new search box for Wikidata that allows an exploratory search based on chained entities without having to navigate through pages. This relies on the classification of data entered by the user in the search box in real-time and the dynamic creation of suggestions through the investigation of entities and relationships.

Since the current methods for determining whether a given declaration exists in Wikidata rely on either using the Wikibase interface, where entities appear as pages containing a list of declarations, or the SPARQL query language used along with the Wikidata Query Service (WDQS), they are complicated for users who are not experts in the field. The paper's idea is built on a web application that has a search engine that can interact with Wikidata while the user types in the search field, thus simplifying the process (using queries being made to the WDQS through SPARQL). Wikidata is asked about its entities when the user starts typing in the search box, allowing the box to provide a list of suggestions with dynamic filtering. The user can view the entities that match their query and choose the one they want by using this list to filter and explore as one can see in Figure 5.[5]

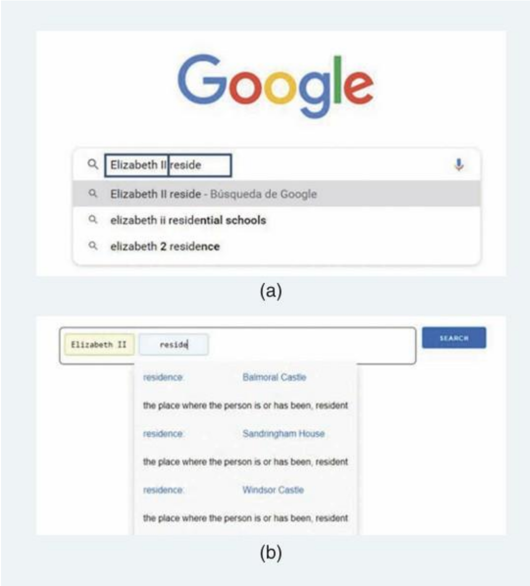

FIG. 5: A search for Elizabeth II's residence, comparing the Google search engine and paper's proposal.

[5]

The two main components of the suggested solution are LiveView, which gathers completed searches and facilitates user exploration of the graph, and the query engine, which sends queries relevant to the user's search terms to Wikidata in order to dynamically obtain information.

To have a better understanding about the application of this feature, paper presents an example with Figure 6 and Figure 7. Figures show a user checking whether Wikidata contains information of the birthplace of Kobe Bryant's wife. By using the proposed search box the user starts by typing "Kobe Bryant", and in real time the interface suggests a list of terms that match the characters, allowing the user to select the requested entity.

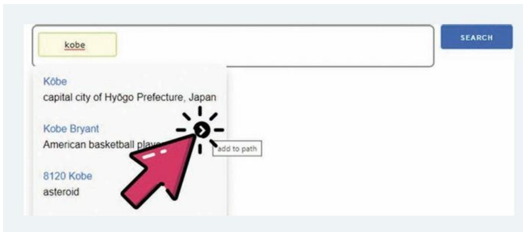

FIG. 6: The initial search and dynamic filtering based on the text introduced by the user.

[5]

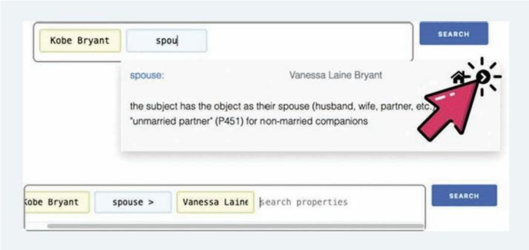

FIG. 7: The search by predicate and dynamic filtering based on a previous entity.

[5]

Moreover, to evaluate the proposal, between the proposed model and Wikidata search a comparison of the complexity of search processes has been done. Each of the 17 queries in the dataset used for this evaluation was created using both tools and included a unique piece of information taken from the 2020 Google Trends for the United States. The analysis is based on two models: an analysis of time and input peripherals (keyboard and mouse), collected with Mouse Miles software and the Firefox add-on Scroll Tracker, and a subset of the most relevant actions of the keyboard-level model (KLM), one can see the results in Figure 8. [5]

Table 1. The keyboard-level model estimation for the 17 elements of the dataset, using the Wikidata search box and the proposed solution.

	Wikidata Search						Prototype Search					
Search	k	p	b	h	m	Total	k	p	b	h	m	Total
S.1	8	2	6	3	8	15.2	10	2	2	3	6	12.8
S.2	5	2	33	3	35	49.7	7	2	2	3	6	12.2
S.3	15	2	3	3	7	15.1	17	2	2	3	6	14.2
S.4	8	2	8	3	12	20.2	11	2	2	3	6	13
S.5	11	3	7	3	12	21.8	17	2	2	3	8	16.6
S.6	9	2	6	3	10	17.8	10	2	2	3	6	12.8
S.7	10	1	6	3	11	18.1	11	1	1	2	5	10.2
S.8	7	3	31	3	35	51	9	3	3	3	7	15
S.9	8	3	7	3	12	21.2	10	2	2	3	6	12.8
S.10	7	2	7	3	11	18.7	9	2	2	3	6	12.6
S.11	8	3	9	3	14	23.8	15	2	2	3	8	16.2
S.12	8	2	7	3	11	18.9	10	2	2	3	6	12.8
S.13	6	2	4	3	8	14.6	8	2	2	3	6	12.4
S.14	9	1	4	3	8	14.1	11	1	1	2	5	10.2
S.15	14	4	16	3	21	35.2	20	3	3	3	9	19.6
S.16	8	8	37	3	33	54.9	14	7	7	3	13	28
S.17	8	1	6	3	11	17.7	10	1	1	2	6	11.2

k: keyboard pulsation; p: pointing to an object or element with the mouse; b: pressing a button on the mouse; h: changing from the mouse to the keyboard (or vice versa); m: mental preparation.
The measurements were obtained using Mouse Miles and Firefox Scroll Tracker.

FIG. 8: The search by predicate and dynamic filtering based on a previous entity.

[5]

As shown in Figure 9, the prototype significantly decreases the amount of time spent on the search process (by 48%), while also significantly increasing the number of keyboard pulsations and significantly reducing the amount of time spent scrolling and moving the mouse. The proposal only required 626 pixels of scrolling in one of the searches, compared to the total 7,278 pixels of scrolling required by all of the Wikidata interface's inquiries. This is a significant reduction in scroll figures.[5]

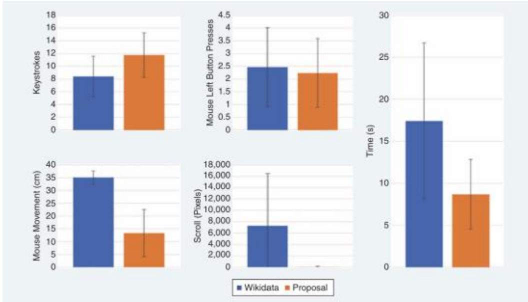

FIG. 9: A comparison of the mean Mouse Miles and Firefox Scroll Tracker results for the 17 dataset elements.

[5]

To fully explore a knowledge graph, a user does not need a technical understanding of SPARQL or other query languages. The findings demonstrate that the suggested approach can greatly enhance Wikidata's search functionality, enabling users to look for specific information more quickly and easily. Based on the KLM results, the prototype is 43% faster than the real Wikidata search system; this tendency is further supported by the 48% time decrease observed in the results from tracking apps used in Firefox. Also, the LiveView module helps to speed up loading times by caching and reducing the amount of information requests made to Wikidata. [5]

D. Knowledge Graph Embeddings & Interactive Visualization Tools

Knowledge graph embeddings represent entities and relations as vectors in a continuous space to capture semantic information and guide exploration. The paper *KGScope: Interactive Visual Exploration of Knowledge Graphs with Embedding-based Guidance* utilizes embeddings for providing three types of guidance for users: peculiarity, relation recommendation, and similar neighbors. The authors of this paper propose an interactive visualization tool called "KGScope" as a solution for visual explorations and provide embedding-based guidance to derive insights from knowledge graphs after they saw the need for more advanced systems that can provide more in-depth analyses most effectively.[4]

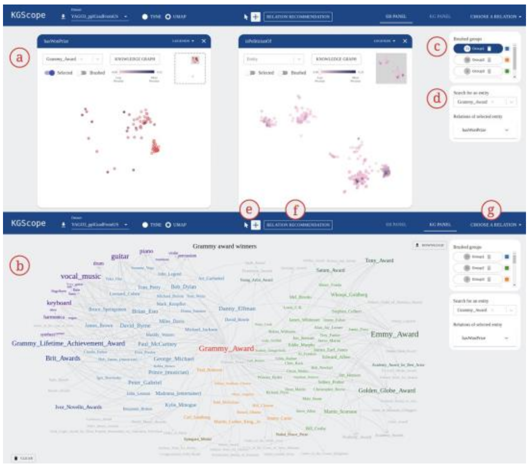

FIG. 10: The interface of KGScope. (a) Embedding panel. (b) Multi-relation knowledge graph panel. (c) Brushed groups panel. (d) Related relation panel. (e) Brush tool. (f) Relation Recommendation button. (g) Schema panel button. In this particular example, the user found that Grammy award winners could be categorized into three groups. Most of the people in Group 1 (blue) have a musical role and have won Grammy Lifetime Achievement Award, Ivor Novello Awards, and Brit Awards, which are all related to music. Many people in Group 2 (green) have won Emmy Awards, Tony Awards, and Golden Globe Awards related to TV, movies, and drama. Group3 (orange) is composed of politicians.
[4]

KGScope provides the context of a knowledge graph by visualizing two types of data:

- the entity-relation schema item visualizes the knowledge graph's structure at a high level, presenting statistics about the relations and entities (authors employ force-directed graphs rather than other graph representations, like the adjacency matrix, to preserve ontological insights), as shown in Figure (11,
- embedding view (EBView) to show the context during a data exploration (authors used the knowledge graph embedding model, TransR, because of its capacity to learn semantic information and model entities in Euclidean space, to facilitate analysis from various context aspects), as shown in Figure (12.

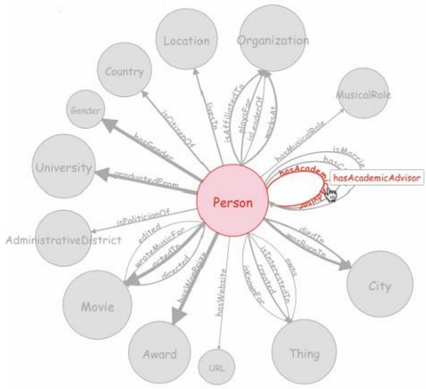

FIG. 11: Hovering on the relation, "hasAcademicAdvisor", in the schema panel of YAGO-US-Graduates.
[4]

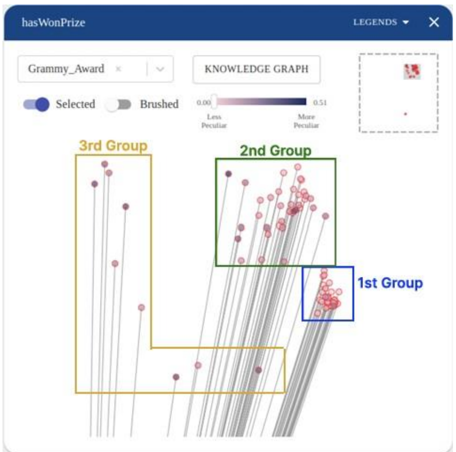

FIG. 12: En example of the distributions of Grammy Award awardees in the EBView.
[4]

Additionally, KGScope offers exploration tools for three different kinds of groups: sibling groups, which are made up of entities connected to a common entity in a relation and can be found in KGView after selecting an entity; visual groups, which are made up of entities grouped by EBView as shown above.

The authors recruited 12 postgraduate students with prior experience in data analysis and administered a post-test questionnaire to them in order to assess the proposed tool. The system they created mimics current knowledge graph exploration systems without embedding-based guidance as the baseline. According to the report, KGScope encourages users to be more open to exploring unfamiliar entities, yielding more surprising finds, enabling users to investigate entities in groups and find intriguing and relevant insights, and generating insights beyond direct links. The comparison between KGScope and Baseline, together with the quantity of insights in each of the four categories, is shown in Figure 13.

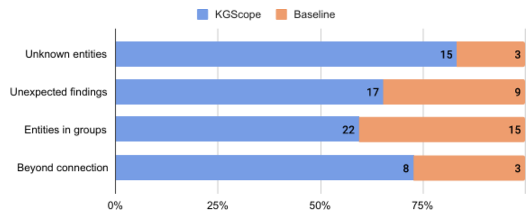

FIG. 13: Comparison of KGScope and Baseline: The number of insights found by participants related to the four categories.

[4]

The findings demonstrate how well KGScope facilitates knowledge graph discovery by offering helpful data and assisting in network investigation.

E. Query Engine and Interaction Flow

Systems enhancing user interaction by simplifying the query process and providing dynamic, real-time search capabilities are crucial.

One of the novel approaches in this area is providing the ability of querying a knowledge graph without having domain expertise. *Example-Driven Exploratory Analytics over Knowledge Graphs* provides Re2xOLAP, an interactive aproach to create OLAP queries from example outputs by reverse engineering.

The system works as follow: Firstly, the system automatically creates a virtual graph by utilizing the access to the triplestore creating that graph, and this graph contains the dimension hierarchy. This graph is held in memory and queries are mapped to this instead. The ReOLAP procedure searches for all dimension members matching the examples in the query separately. **Complexity:** For each member of the user example, running time grows proportionally to the number of dimensions, the number of hierarchies, and the cardinality of their members, but time complexity is independent of the actual number of observations, thanks to virtual graph based matching.

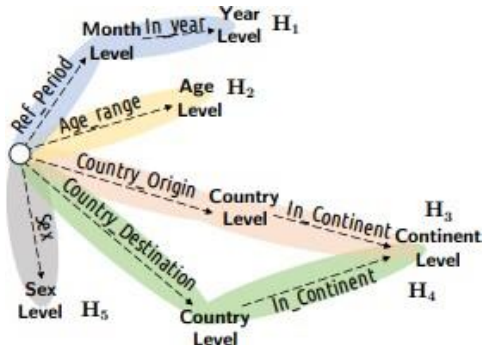

FIG. 14: Virtual Graph for Requests of Asylum data

[8]

After a query presented to the user, they move on to refinement phrase where user can choose how to proceed: disaggregate, subset, and find similar.

An example workflow of this algorithm is shown in Figure 15

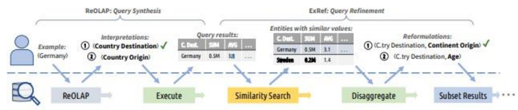

FIG. 15: Re2xOLAP: example query synthesis and refinement steps, an interaction step is mapped to a pair of arrows

[8]

Creation of the virtual schema containing the hierarchy relations inside the model requires system bootstrapping.This operation is done only once offline and any update only on the data side is reflected highly fast with in memory operations. Running time varies from 25 minutes for DBpedia up to 60 minutes for Eurostat. They note that the bootstrap time is influenced mainly by the complexity of the schema and not by the number of observations.[8]

In conclusion, Re2xOLAP is the first solution providing example driven exploratory analysis on statistical KGs, mitigating the need to formulate a query in SPARQL.

F. Progressive and Incremental Results

Progressive and incremental results techniques provide initial results quickly and refine them progressively as more data is processed.

AttnIO: Knowledge Graph Exploration with In- and-Out Attention Flow for Knowledge-Grounded Dialogue tries to implement a progressive strategy to choose the range of entitites to attend, depending on the characteristics of a given dialog.

The model trained in the provided research paper is called AttnIO. It utilizes a Graph Neural Network (GNN) architecture incorporating a message passing mechanism typical in GNNs. AttnIO employs attention mechanisms to facilitate navigation through knowledge paths within a knowledge graph. This approach ensures the model's resilience to less-than-optimal paths, thereby enhancing the accuracy of tasks related to knowledge retrieval.

In contrast to recurrent decoder-based models like DialKG Walker and Seq2Path, which rely on a single state vector and may struggle to recover from sub-optimal choices, AttnIO differs by distributing the decoder state across feature vectors of all navigable entities.

During each decoding step, initial attention values are computed based on how relevant candidate nodes are to the dialog context, determined by the dot-product of node features and the context vector. As decoding progresses, the model dynamically updates its attention distribution over the graph, effectively integrating dialog context with the structural knowledge encoded in the knowledge graph. [7]

Dialog	A: *Fiona Stafford wrote Emma. It's a romance novel. Are you into that genre?* B: *Any other books that might fall under comedy? I'm in the mood for something light.* A: *[RESPONSE]*
AttnIO-AS	*Comedy ⇒ subject of ⇒ The War of the Worlds ⇒ written by ⇒ Arthur. C. Clarke*
AttnIO-TS	*Comedy ⇒ subject of ⇒ The War of the Worlds ⇒ subject ⇒ Comedy*
AttnFlow	*Comedy ⇒ parent genre ⇒ Slapstick*
GT	*Comedy ⇒ subject of ⇒ One Crazy Summer*

FIG. 16: Sample paths generated from each model, along with the ground-truth path.
[7]

Figure 16 showcases a path generation of AttnIO algorithm given a dialog sample.
Table VI shows that the model trained with all path supervision (AttnIO-AS) significantly outperforms all baselines. In conclusion AttnIO provides a versatile model that can be trained on any context.

IV. DISCUSSION

The reviewed papers collectively propose a comprehensive approach to enhance KG exploration. By combining advanced summarization metrics, user-friendly interfaces, interactive visualizations,

TABLE V: Recall@k for different models

Model	path@5	path@10	tgt@10
Seq2Seq	29.7	44.1	-
Tri-LSTM	22.6	36.3	-
EXT-ED	9.0	13.3	-
DialKG Walker	35.3	47.9	-
Seq2Path	31.1	38.68	42.52
AttnFlow	30.68	39.48	58.84
AttnIO (GRU)	42.98	51.22	64.33
AttnIO (no context)	31.03	40.39	58.32
AttnIO (no alignment)	41.19	48.85	64.01
AttnIO-AS	43.57	52.17	65.48
AttnIO-TS	30.5	39.48	61.04

TABLE VI: Performance of AttnIO in OpenDialKG, in comparison with baselines and ablation models

[7]

advanced machine learning algorithms and query engineering. These solutions improve the accuracy of exploration results and user experience by reducing the effort required for exploration in different ways.

Our state-of-art clearly shows that knowledge graph exploration can take various different forms that are benefitial to different users and use cases. Our research question throughout this literature reading has been "How can we improve the accessibility and effectiveness of Knowledge Graph exploration for both expert and non-expert users?". In that regard, to our best knowledge, the proposed papers are the most novel approaches in their field and their algorithmic experiment results show the improvement they made on state-of-art in their niche. Each of them touch on this research question either on expert or non-expert user side, improving one aspect.

Moreover, most of these papers include the missing aspects in their research and methodologies, providing guidance for future research for interested researchers. We highly suggest reading more in-depth on the paper you are interested in.

V. CONCLUSION

Knowledge Graph Exploration is crucial for understanding and utilizing KGs effectively. This report has reviewed the state-of-the-art in KG exploration, highlighting key tasks, challenges, and recent advancements. By addressing issues of accessibility, effectiveness, and scalability, the proposed methods and techniques significantly enhance the exploration process for both expert and non-expert users.

[1] M. Al-Tawil, V. Dimitrova, D. Thakker, and B. Abu-Salih, ”Emerging Exploration Strategies of Knowledge Graphs,” in *IEEE Access*, vol. 11, pp. 94713-94731, 2023, doi: 10.1109/ACCESS.2023.3308514. keywords: Task analysis;Knowledge graphs;Resource description framework;Instruments;Music;Navigation;Data visualization;Semantic Web;Knowledge graphs;knowledge utility;exploration experience;exploration paths;exploration strategies.

[2] Lissandrini, M., Mottin, D., Hose, K., & Pedersen, T. B. (2022). Knowledge Graph Exploration Systems: Are We Lost? CIDR 2022. Link

[3] Wang, X., & Cheng, G. (2023). A Survey on Extractive Knowledge Graph Summarization: Applications, Approaches, Evaluation, and Future Directions. Link

[4] Yuan, H., Yu, T.-W., Pan, J.-Y., & Lin, W.-C. (2021). KGScope: Interactive Visual Exploration of Knowledge Graphs with Embedding-based Guidance. IEEE Transactions on Visualization and Computer Graphics, 14(8), 1-12. Link

[5] Yagüe París, J., Espada, J. P., Martínez, J. S., & Crespo, R. G. (2024). Search Box System to Improve User Interaction in Knowledge Graph Searches: A Solution for Users Without Expert Skills. IEEE Systems, Man, & Cybernetics Magazine. Link

[6] Pappas, A., Troullinou, G., Roussakis, G., Kondylakis, H., Plexousakis, D. (2017). Exploring Importance Measures for Summarizing RDF/S KBs. In: Blomqvist, E., Maynard, D., Gangemi, A., Hoekstra, R., Hitzler, P., Hartig, O. (eds) The Semantic Web. ESWC 2017. Lecture Notes in Computer Science(), vol 10249. Springer, Cham. Link

[7] Jung, S., Jang, Y., Kim, J., Kim, K., Hwang, W., & Song, Y. (2020). AttnIO: Knowledge Graph Exploration with In-and-Out Attention Flow for Knowledge-Grounded Dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Retrieved from Link

[8] Matteo Lissandrini, Katja Hose, and Torben Bach Pedersen, “Example-Driven Exploratory Analytics over Knowledge Graphs,” *Proceedings of the 26th International Conference on Extending Database Technology, EDBT 2023*, 2023. Link

[9] D. Wu, M. Tang, S. Zhang, et al., *KPRLN: deep knowledge preference-aware reinforcement learning network for recommendation*, *Complex Intell. Syst.*, vol. 9, pp. 6645–6659, 2023. Link

[10] Chen, Y., Min, S., Chen, D., Wang, M., & Ren, X. (2020). Improving Commonsense Question Answering by Graph-based Iterative Retrieval over Multiple Knowledge Sources. In *Proceedings of COLING 2020* (pp. 2633-2646). Association for Computational Linguistics. Retrieved from Link

[11] G. Troullinou, H. Kondylakis, E. Daskalaki, and D. Plexousakis, *Ontology understanding without tears: the summarization approach*, Semantic Web Journal, 2017, IOS Press.