



KvGR: A Graph-Based Interface for Explorative Sequential Question Answering on Heterogeneous Information Sources

Hans Friedrich Witschel^(✉), Kaspar Riesen, and Loris Grether

FHNW University of Applied Sciences and Arts Northwestern Switzerland,
4600 Olten, Switzerland

{hansfriedrich.witschel,kaspar.riesen,loris.grether}@fhnw.ch

Abstract. Exploring a knowledge base is often an iterative process: initially vague information needs are refined by interaction. We propose a novel approach for such interaction that supports sequential question answering (SQA) on knowledge graphs. As opposed to previous work, we focus on exploratory settings, which we support with a visual representation of graph structures, helping users to better understand relationships. In addition, our approach keeps track of context – an important challenge in SQA – by allowing users to make their focus explicit via subgraph selection. Our results show that the interaction principle is either understood immediately or picked up very quickly – and that the possibility of exploring the information space iteratively is appreciated.

Keywords: Sequential question answering · Graph databases · Natural language interfaces

1 Introduction

Today’s information repositories are numerous, diverse and often very large. There is an increasing demand for accessing and querying these repositories using questions posed in natural language. While there is a long history of research in the fields of Question Answering (over both structured and unstructured content) and Natural Language Interfaces to Databases (NLIDB), as further elaborated in Sect. 2, the field of (*Complex*) *Sequential Question Answering* [5, 14] is still rather new.

Possibly fuelled by the rise of chatbot technology and the resulting expectations of users, it claims that a more interactive approach to both fields will better meet user needs. Its main assumption is that users do not simply ask a question to a knowledge base and then quit. Instead, users tend to break down complex questions into a series of simple questions [5]. In addition, as known from exploratory search [12], users who do not have a very clearly articulated information need and/or who aim at getting familiar with a new field of knowledge tend to ask series of questions where one answer triggers the next question. That is, a user might ask a rather “fuzzy” first question (such as “what are

important topics in the field of ‘Information Retrieval’?) and then – when studying the answer – start to think of new questions, concerning some of the new concepts found in that answer. Although the concept of exploratory search is well known from the field of information retrieval, this exploratory motivation for performing sequential question answering (over structured knowledge bases) has not been studied so far. In any case, sequential question answering raises the major challenge of keeping track of context: since they assume the context to be known from the prior questions and answers, users tend to leave away sentence elements [14].

Especially in exploratory search settings, answers to fuzzy questions can be very complex, involving a large number of concepts and relations. Hence, researchers have proposed various kinds of visualisations in order to aid users in grasping such complexity and studying relationships between concepts [1,20].

In our work, we aim at building a context-aware sequential question answering system, especially suited for exploratory search. To this end, the solution is based on a knowledge graph – which integrates information from various structured and unstructured data sources, see Sect. 3.1. Since the visualization of graphs provides an intuitive overview of complex structures and relationships [2], our system allows users to ask questions in natural language, but provides answers via a visual representation of subgraphs of the underlying knowledge graph. It supports both the user and the system in keeping track of the context/current focus of the search via a novel interaction concept that combines pointing/clicking and asking questions in natural language, described in Sect. 3.2.

We will show empirically that users appreciate the new interaction concept and its ability to define context and focus graphically, see Sect. 4.

2 Related Work

Both question answering and natural language interfaces to databases (NLIDB, see [6] for a survey) have a long history. They share many characteristics since both support querying of knowledge bases using natural language. Many question answering systems retrieve answers from textual (i.e. unstructured) resources, but there are also many approaches based on structured content, often in the form of ontologies [11].

In NLIDB, many challenges have been addressed, e.g. making systems domain-independent [10] or overcoming specific difficulties with certain query languages, above all SQL [22]. Recent advances in this area are relying on sequence-to-sequence models [7,17], based on encoding and decoding of sequences via deep (reinforcement) learning. An obvious drawback of these supervised learning approaches – as opposed to earlier hand-crafted rule-based grammars – is the amount of training data required. Although large hand-annotated datasets have been published [24,25], trained models cannot be expected to be fully domain-independent.

While the fields of Question Answering (over structured data), Semantic Parsing and NLIDB are obviously quite advanced, researchers have only recently begun to study the domain of “Sequential Question Answering” (SQA). This new focus on interactive, dialog-driven access to knowledge bases is based on the insight that users rarely pose a question to such a knowledge base and then quit [3, 14]. Instead, a more common and natural access pattern consists in posing a series of questions. Most researchers in SQA assume that the motivation for dialogs comes from the need to decompose complex questions into simple ones [5, 14]. Some researchers propose to perform such decomposition algorithmically [15], while others provide evidence that it is more natural and realistic to assume that humans will like to perform this decomposition themselves, resulting in a series of simple, but inter-related questions [5]. A key challenge in any form of sequential or conversational question answering is the resolution of ellipses (e.g. omissions or arguments in relations) or anaphora which are very frequent in a dialogue where the user expects the system to keep track of the context [5, 9, 14].

These approaches all assume that a searcher always accesses a knowledge base with a clear question in mind. As outlined above, we advocate a wider perspective on SQA, including scenarios of an exploratory nature. In information retrieval, it has been thoroughly accepted that there exist situations in which users are unable to clearly articulate information needs, e.g. when trying to get acquainted with a new field where terminology is still unknown [12]. Thus, users would like to *explore*, and often their questions become better articulated as they learn more about the new field.

In order to support them in grasping relationships between new concepts in the – often very complex – answers to their fuzzy questions, IR researchers have proposed result set visualisations that provide a better overview than the typical ranked lists of document references [1, 20].

Using visualisations, especially of graphs/ontologies as an output of retrieval systems has also been proposed, mainly in QA and NLIDB that are based on knowledge graphs [2, 13, 23].

Visualising graph query results is different from visualising graphs in general; the former resembles generation of results snippets in text retrieval [16]. However, we can learn and employ mechanisms from general approaches to analysing large graphs, e.g. by applying global ranking mechanisms (such as PageRank) or by summarizing properties of selected nodes [8]. As pointed out in [19], visual graph analysis requires, besides the visual representation of graph structures, to have good interaction mechanisms and algorithmic analysis, such as aggregation/merging of nodes, identification of certain graph structures (such as cliques) or node ranking mechanisms such as PageRank.

Additional challenges originate in the fuzziness of natural language and the potential resulting number of (partially) matching result graphs. Graph summarization approaches have been proposed as a solution [21, 23] – where summarized/aggregated graph structures play the role of snippets. Another approach [4] uses result previews to narrow down result sets via “early” user interaction.

2.1 Contribution

While approaches to semantic parsing, NLIDB and question answering over structured data are well studied, there is a recent rise in interest in better studying and supporting the interaction in sequential question answering (SQA) scenarios.

However, the emerging field of SQA lacks – in our opinion – a clear idea of why users want to engage in a conversation. We claim that one important motivation can be found in exploratory settings where users need to first gain insights by interacting with a knowledge base, before being able to ask the “right” questions. Another challenge in SQA is keeping track of context: in their survey on semantic parsing, Kamath & Das [6] mention “adding human in the loop for query refinement” as a promising future research direction in cases where the system is uncertain in its predictions.

Our contribution consists mainly in proposing a new interaction paradigm which allows users to ask questions in natural language and to receive answers in the form of visualised subgraphs of a knowledge graph. Users can then interact with that subgraph to define the focus of their further research, before asking the next question. With this human involvement, we can show empirically both how the human benefits from clarifying the search direction while exploring the knowledge graph and how the machine is supported in understanding incomplete questions better because their context is made explicit.

We further use a robust query relaxation approach to trade precision for recall when recall is low. Our approach is domain-independent and does not require training data – it only requires a specification of node type names and their possible synonyms. It can be seen as a “traditional” and simple grammar-based approach – the focus is not on sophisticated semantic parsing (we might add e.g. sequence-to-sequence models later), but on the interactive process of graph exploration via natural language.

3 The Retrieval System

3.1 Graph-Based Integration of Heterogeneous Information Sources

The knowledge graph underlying our experiments was constructed out of a collection of heterogeneous sources and stored in a Neo4j graph database¹. For our experiments, we chose books as a domain and aimed at retrieving all information – from various sources – which users (leisure-time readers, students, ...) might find relevant, ranging from core bibliographic information, over author-related information (affiliation/prizes won) to reviews and social media coverage of books.

To populate it, we implemented a collection of parsers for a variety of data sources.²:

¹ <https://neo4j.com/>.

² Note: these are meant to illustrate different *types* of data, the concrete sources do not matter in general.

- For *structured data*, we built an XML parser (which can be applied to structured XML databases, but also for semi-structured XML files) and an RDF parser. The XML parser was used to integrate a sample of data from the bibliographic platform iPEGMA³, while the RDF parser was applied to the DBPedia SPARQL endpoint⁴ to retrieve data about books, persons, their institutes and awards. The iPEGMA data covers mostly German books while DBPedia data is focused on English books.
- In terms of *semi-structured data*, our HTML parser can process web content and a special Twitter parser deals with Tweets (and uses the HTML parser to process web pages linked from tweets). We applied the HTML parser to the websites literaturkritik.de and www.complete-review.com to retrieve book reviews and related book metadata in German and English. The Twitter parser was applied to a collection of Twitter accounts of major publishers whose timelines were analysed for tweets referring to books.
- We also integrated a sentiment analysis service (Aylien Text API⁵) as a typical example of analysis of the *unstructured* part of webpages, i.e. the plain text. In our case, we applied the service to the book reviews from literaturkritik.de to find out whether reviews were positive or negative. For www.complete-review.com, this information could be parsed directly from the web page.

In Neo4j, it is not required to define a schema (i.e. node or relation types) before inserting nodes or relationships. We used this property heavily: each parser has a configuration file in which one can define node and relation types to be extracted. We have developed a special syntax with which one can define the patterns to be searched within the various data sources to retrieve the corresponding data. This means that parsers can be extended to find new types of nodes and relationships and/or cover new data sources of known type, without the need to modify the program code of the parser. Typically, the specifications for various data sources have overlapping node types, thus resulting in a data integration task. In order to match identical nodes (e.g. the same book) found in different data sources, the definitions also specify a “uniqueness attribute” (similar to a primary key in relational databases). As a result, the knowledge base consists of a single integrated graph.

We have chosen a graph database because graphs are a very natural way of modeling relationships and are easy to visualise and interact with [2].

3.2 The Interaction Concept

As laid out in Sect. 2, most previous work sees sequential question answering as a conversation in which complex questions are broken down into simpler ones. For instance, Iyyer et al. [5] assume that users have already at the initial state

³ <https://ipegma.com/>.

⁴ <http://dbpedia.org/sparql>.

⁵ <https://aylien.com/text-api/>.

of a conversation a complex question in mind – which they then decompose into simpler ones.

In contrast, our new interaction concept aims at supporting scenarios that are more *exploratory* in nature (cf. exploratory search in text retrieval [12]). In such settings, users often ask series of questions that emerge one from another – i.e. the answer to a first question triggers the next one etc. – without the final goal of such a conversation being clear initially.

We propose a novel interaction mechanism for such an exploratory “conversation”, where questions are posed in natural language, but answers are given in the form of subgraph visualisations, with a possibility to interact and select parts of subgraphs for further exploration (again via asking questions). Note that it does not play a role whether a user starts from general concepts to “zoom in” to more specific ones or vice versa.

In exploratory search, it is typical that – since the nature of the problem is unclear to the user – queries are imprecise or “tentative” [20]. This implies very often that the *answers* – much more than the questions or queries – can be quite complex. As pointed out in [1], systems that support exploration hence often offer visualisation of search results as well as interaction mechanisms for further exploration.

In our case, results are (possibly large) subgraphs of a given knowledge graph. By studying such a subgraph and interacting with it, a user may *learn* about important concepts and relations in a domain – and this leads to asking the next question(s). A next question may aim at either *filtering* the current subgraph or further broadening the scope by *expanding* a subgraph region with further related nodes.

The design of our interaction concept was informed by a questionnaire which was filled out by a sample of 16 students. Participants received a description of a situation (e.g. having read a good book) and were asked to formulate some questions that they would have in such a situation. We analysed their answers, looking for common patterns of questions and expected result sets.

Our resulting interaction concept is very simple: based on an initial keyword search or question in step 0, a user finds an entry point into the graph, i.e. an initial subgraph G_0 .

From this point on – provided that the user would like to continue the current session – there are two main possibilities for exploration in each step i :

1. Use the graphical user interface, e.g. expand the subgraph G_i by unhiding all nodes related to a chosen node.
2. Select a node or a set of nodes N_{G_i} as a “context” and ask a question about it. Selection can be done
 - (a) directly via one or more clicks on nodes or
 - (b) by selecting all nodes of a certain type via a button.

Each interaction leads to a new graph G_{i+1} .

While option 1 is not new, option 2 can lead to a new form of sequential question answering, with questions being asked in natural language and answers

given as visualisations of subgraphs. This combination is user-friendly since on the one hand – as a basis of all NLIDB research and conversational interfaces – natural language is the most natural form of expressing information needs. On the other hand, researchers in both information retrieval [1] and graph querying [23] communities use visualisations for improving the user-friendliness of exploratory search.

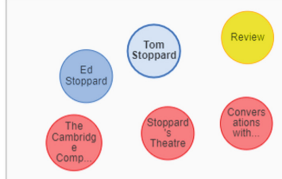


Keywords / question	Result	Chosen context
«Stoppard»	 <p>Legend: Person (blue), Review (yellow), Book (red). Nodes: Tom Stoppard, Ed Stoppard, The Cambridgeshire Comp., Stoppard's Theatre, Conversations with, Review.</p>	Tom Stoppard (2a)
«Show me all books by this author!» (<i>expansion</i>)	 <p>Legend: Person (blue), Review (yellow), Book (red). Nodes: Tom Stoppard, Hanggood, Shipwreck, Rock 'n' Roll, Voyage, Night and Day, Savage, Professional Foul, Jumpers, The Rover.</p>	Book (all red nodes, 2b)
«Which of these books have positive reviews?» (<i>filter</i>)	 <p>Legend: Person (blue), Review (yellow), Book (red). Nodes: Reviews, various books.</p>	

Fig. 1. An exemplary conversation between a user and KvGR (Color figure online)

In addition, we claim (and will later show empirically) that, while it is not natural for users to repeat entity names from an earlier question, it is rather natural for them to select preliminary results and thus make context explicit. We will show that such selection is even often helpful for their own understanding of how a question-answer-sequence develops and what they have learned so far/what they want to learn next.

Since the user specifies the context explicitly when using option 2, it is easy for our system to fill in missing parts of questions by simply assuming that they originate from that context.

Figure 1 illustrates the interaction concept with a small “exploration session” from the book domain (see Sect. 3.1). In short, the session consists in a user searching for an author, then demanding to see all books from that author and finally asking which of these books have positive reviews. Note how the visualisation of the result graph helps her to get a quick overview of complex structures – for instance to see at a glance which books have many vs. few positive reviews (yellow nodes) in the last result.

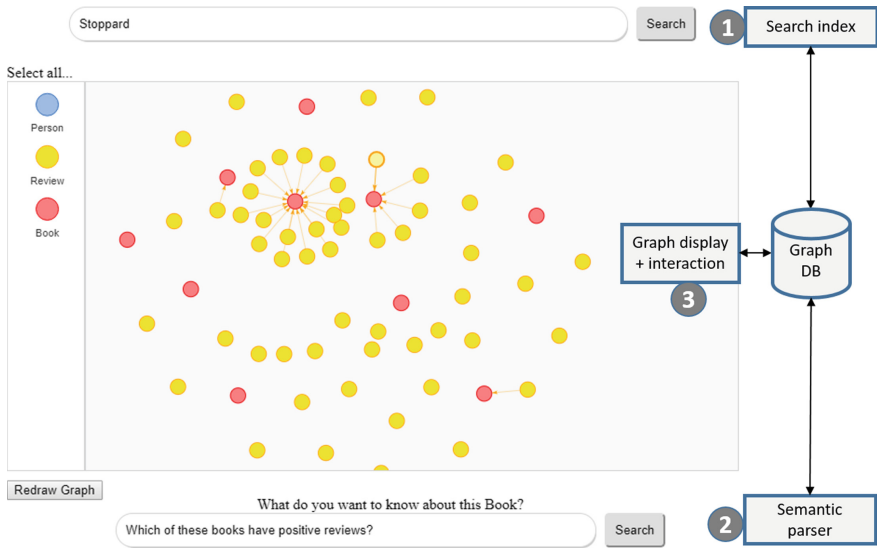


Fig. 2. A screenshot of the KvGR core UI components

3.3 The KvGR Architecture

In order to realise the interaction described in the previous section, KvGR builds several components on top of the knowledge graph (see Sect. 3.1). All of these components are visible on the user interface, the numbers in Fig. 2 refer to the corresponding (backend) components in the following enumeration:

1. **Fielded keyword search:** each node in the knowledge graph is treated as a document and its (textual) attributes as fields. Field weights are domain-specific – in the book domain the “title” field of books will have a higher weight than e.g. the “genre” field. The number of shown nodes is limited by applying a cut-off to node scores.
2. **Semantic parser,** see Sect. 3.4
3. **Graph visualisation and interaction,** allowing common basic graph interactions, plus selecting a context, see Sect. 3.2.

3.4 Semantic Parser

Since semantic parsing is not the core contribution of our work, we have built a simple, but robust grammar for parsing. It takes advantage of the interaction concept and the basic principles of graphs, but makes no further assumptions about the graph schema – it can be adapted easily to new domains simply by providing a lexicon of node types (see below).

The grammar consists of JAPE rules in GATE⁶, which annotate occurrences of graph nodes in user utterances, based on a simple lookup mechanism using a lexicon with manually maintained synonyms. Each annotation is associated with a number of features, see Fig. 3.

The annotated questions are then passed to a Cypher generator, which simply takes all nodes found in an utterance and generates a relationship pattern that is matched against the graph.

We illustrate our parser with the example shown in Fig. 3.



Fig. 3. A user utterance, with annotated nodes

The parts of the question recognised as nodes are put in bold font, their extracted features are presented in the box above. The grammar has marked “journal” as a return node type and “it” as referring to a current user selection (“this=true”). Here, the interaction concept is exploited: because the user has selected a book (let us assume, the book with id 629025), the system can assume that the pronoun “it” refers to that current selection (the same would apply to a phrase like “this book”).

This information is enough for the Cypher generator to generate a Cypher query as follows:

```
match (x:Journal)-[]-(y:Book) where ID(y) IN [629025] return x,y
```

This query, however, will not retrieve anything since the question contains an ellipsis: it should actually be formulated as “Which **journals** have published a **review** about **it**?”. That is, the system needs to extend the pattern to allow an intermediate node type related to both the current selection and the return type nodes.

To this end, we have implemented a *query relaxation mechanism* which will first try out the above Cypher query and then – if nothing is returned – will relax the query by allowing an ellipsis like this:

```
match (x:Journal)-[]-(z)-[]-(y:Book) where ID(y) IN [629025] return x,y,z
```

⁶ <https://gate.ac.uk/sale/thakker-jape-tutorial/GATE%20JAPE%20manual.pdf>.

The system does not know/specify that the intermediate node z is of type Review – thus a negative impact on retrieval precision might result, which we trade for recall here.

4 Evaluation and Discussion

In order to evaluate our main hypothesis – namely that our new interaction mechanism effectively supports users in iteratively refining an exploratory search – we performed user tests in an exploratory search scenario.

To make the sessions more comparable, we pre-defined the information needs: the “story” started with a keyword search for the topic “criminal law” and was continued with some typical questions about e.g. prominent authors in that field, authors who had won prizes, their institutes, as well as books with positive reviews in that field. Before each session, participants were instructed about the features of the system via a short demo. Within the session, the predefined information needs were explained and users were asked to interact with the system to satisfy them. When users got stuck with interaction or query formulation, help was offered.

Following the popular “five-user assumption” of usability testing [18], we recruited 5 participants, 2 colleagues from our School of Business and 3 of our students. All subjects were not previously aware of our project. This selection was made for practical feasibility reasons – we are aware of the bias, in terms of user characteristics, that it introduces.

4.1 Observations

Participants received overall 5 different information needs (q_1 to q_5). The first one (q_1) started from a single node (the topic “criminal law”), i.e. a context selection was not required. All subsequent ones required participants to select a subset of the nodes that were currently displayed (e.g. all books or all persons). The last information need (q_5) was formulated in a complex way (“which authors that have written a book about criminal law have also written a review?”) and required participants to recognise that a partial result to the question was already available from a previous step.

We observed the participants’ difficulties to (a) formulate queries that the semantic parser would understand correctly, (b) grasp the principle of breaking down complex information needs into simpler ones (here, participants would typically try to extend the previous query by adding more constraints) and to (c) remember to select a subset of nodes as a context for their next query.

Table 1 shows the number of participants facing these problems for each of the test queries. In terms of query reformulation, there is no clear pattern – we observed a number of ways in which our grammar can be improved.

Grasping the process of iterative refinement shows a clear learning curve: while two participants had understood the principle immediately from the introductory demo, the other three needed only one experience with q_2 to grasp it.

Table 1. Number of test persons encountering problems for each test query

Type of problem	q_1	q_2	q_3	q_4	q_5
Reformulation required	1	5	3	0	2
Struggle with breakdown	–	3	0	0	2
No context selection	–	3	2	1	1

We observed that the problems with q_5 resulted merely from participants not accurately understanding the complex question – they both said that it would have been different if it had been their own information need.

Remembering to select a subset of nodes as a context was harder: while two participants never forgot to do it, one needed q_2 , another one q_2 and q_3 to remember it; one participant could not get used to it until the end of the test. The persons who struggled expressed their expectation that – if they did not select any nodes, but asked a question like “which of these persons...” – the system should automatically assume that it referred to all currently visible persons. Since this is easy to build into our system, we can conclude that context selection will not be an issue once the principle of iterative refinement has been grasped.

4.2 Feedback

Besides observing the query formulation and interaction strategies of the users – including their need for help – we asked the users to give us feedback on the following points:

- **Intuitiveness of context selection:** three participants stated that they found it intuitive and natural to select a context for their query and to break down complex questions. The other two expressed their expectation for the system to identify context automatically (see above).
- **Results of elliptic queries:** queries containing “intermediate nodes”, e.g. a query “show me all authors who have written about criminal law” would show not only authors, but also their books, although the question did not ask for books. Only one participant had difficulties in understanding what was shown (because the legend was not clear to him). When judging the result, 4 participants said that seeing the books was interesting, especially for someone wishing to explore criminal law as a new area, while 3 participants remarked that the result was not strictly what they had asked for. Two participants stated that they would appreciate to see a list of persons – in addition to the graph visualisation.
- **General feedback** on the interaction was very positive. Despite the observed difficulties that did occur with query formulation, all participants said that they were impressed with the ability of the system to understand queries in natural language. Four participants mentioned explicitly that the visual representation helped them to better understand relationships and to see “how

things belong together”. One participant said that it sparked his curiosity to explore further. All participants stated that the interaction mechanism was either “intuitive” or at least “easy to learn” (because, as they stated, “the effect of what you do is always visible”) and three of them mentioned expressly that they liked the refinement process of breaking down complex queries.

Participants also came forth with a number of suggestions for improvement: two participants stated that they would appreciate if the system could understand – besides fully formulated questions – keyword-based inputs. The same participants and a third one expressed their wish to have result lists, in addition to a graph. The main reason mentioned for this was the lack of a ranking provided in the graph. The participants said that they would not know where to start looking if a result graph grew too large.

- **Comparison to traditional interfaces**, especially ones with list-based result presentation: participants said that our system would be more effective in supporting “detailed investigation” that required to “understand relationships”, whereas traditional list-based systems would be better suited to get an overview of e.g. the most important books on criminal law because of their clear ranking.

5 Conclusions

In this work, we have proposed a novel context-aware sequential question answering system, especially suited for exploratory search, based on graph visualisation for result presentation and iterative refinement of information needs. This refinement in turn is based on the selection of subsets of nodes for context definition and natural language questions towards this context.

Our results are somewhat limited by the specific scenario and use case that we explored and the small user group involved. However, they do show quite clearly that users either understand the principle immediately or pick it up very quickly – and that they appreciate the possibility of exploring the information space iteratively. Having to explicitly select context is hard to get used to for some, and should be automated. The visual representation of results was well received for its support of understanding relationships. On the other hand, it became clear that ranking or highlighting the more “relevant” nodes will be needed to help users focus, especially when results get larger.

Thus, our main goal for future work will be to investigate the best way to incorporate node scoring into the system – either visually (e.g. via node sizes) or by providing ranked result lists in addition to and linked to the graph. Because of the limitations of our participant selection strategy, further test with a more varied user group will also be required. Finally, it might be interesting to explore the possibility for users to combine search results (sub-graphs) of queries before exploring the combined results further.

References

1. Ahn, J.W., Brusilovsky, P.: Adaptive visualization for exploratory information retrieval. *Inf. Process. Manag.* **49**(5), 1139–1164 (2013)
2. Bhowmick, S.S., Choi, B., Li, C.: Graph querying meets hci: state of the art and future directions. In: *Proceedings of the 2017 ACM International Conference on Management of Data*, pp. 1731–1736. ACM (2017)
3. Guo, D., Tang, D., Duan, N., Zhou, M., Yin, J.: Dialog-to-action: conversational question answering over a large-scale knowledge base. In: *Advances in Neural Information Processing Systems*, pp. 2942–2951 (2018)
4. Hung, H.H., Bhowmick, S.S., Truong, B.Q., Choi, B., Zhou, S.: QUBLE: towards blending interactive visual subgraph search queries on large networks. *VLDB J. Int. J. Very Large Data Bases* **23**(3), 401–426 (2014). <https://doi.org/10.1007/s00778-013-0322-1>
5. Iyyer, M., Yih, W.T., Chang, M.W.: Search-based neural structured learning for sequential question answering. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1821–1831 (2017)
6. Kamath, A., Das, R.: A survey on semantic parsing. arXiv preprint [arXiv:1812.00978](https://arxiv.org/abs/1812.00978) (2018)
7. Keneshloo, Y., Shi, T., Ramakrishnan, N., Reddy, C.K.: Deep reinforcement learning for sequence to sequence models. arXiv preprint [arXiv:1805.09461](https://arxiv.org/abs/1805.09461) (2018)
8. Koutra, D., Jin, D., Ning, Y., Faloutsos, C.: Perseus: an interactive large-scale graph mining and visualization tool. *Proc. VLDB Endowment* **8**(12), 1924–1927 (2015)
9. Kumar, V., Joshi, S.: Incomplete follow-up question resolution using retrieval based sequence to sequence learning. In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 705–714. ACM (2017)
10. Llopis, M., Ferrández, A.: How to make a natural language interface to query databases accessible to everyone: an example. *Comput. Stand. Interfaces* **35**(5), 470–481 (2013)
11. Lopez, V., Uren, V., Sabou, M., Motta, E.: Is question answering fit for the semantic web?: a survey. *Seman. Web* **2**(2), 125–155 (2011)
12. Marchionini, G.: Exploratory search: from finding to understanding. *Commun. ACM* **49**(4), 41–46 (2006)
13. Park, C.S., Lim, S.: Efficient processing of keyword queries over graph databases for finding effective answers. *Inf. Process. Manag.* **51**(1), 42–57 (2015)
14. Saha, A., Pahuja, V., Khapra, M.M., Sankaranarayanan, K., Chandar, S.: Complex sequential question answering: towards learning to converse over linked question answer pairs with a knowledge graph. In: *Thirty-Second AAAI Conference on Artificial Intelligence* (2018)
15. Talmor, A., Berant, J.: The web as a knowledge-base for answering complex questions. arXiv preprint [arXiv:1803.06643](https://arxiv.org/abs/1803.06643) (2018)
16. Turpin, A., Tsegay, Y., Hawking, D., Williams, H.E.: Fast generation of result snippets in web search. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 127–134. ACM (2007)
17. Utama, P., et al.: An end-to-end neural natural language interface for databases. arXiv preprint [arXiv:1804.00401](https://arxiv.org/abs/1804.00401) (2018)

18. Virzi, R.: Refining the test phase of usability evaluation: how many subjects is enough. *Hum. Factors* **34**(4), 457–468 (1992)
19. Von Landesberger, T., et al.: Visual analysis of large graphs: state-of-the-art and future research challenges. *Comput. Graph. Forum* **30**, 1719–1749 (2011). Wiley Online Library
20. White, R.W., Kules, B., Bederson, B.: Exploratory search interfaces: categorization, clustering and beyond: report on the XSI 2005 workshop at the Human-Computer Interaction Laboratory, University of Maryland. In: *ACM SIGIR Forum*, vol. 39, pp. 52–56. ACM (2005)
21. Wu, Y., Yang, S., Srivatsa, M., Iyengar, A., Yan, X.: Summarizing answer graphs induced by keyword queries. *Proc. VLDB Endowment* **6**(14), 1774–1785 (2013)
22. Xu, X., Liu, C., Song, D.: SQLNet: generating structured queries from natural language without reinforcement learning. arXiv preprint [arXiv:1711.04436](https://arxiv.org/abs/1711.04436) (2017)
23. Yang, S., et al.: SLQ: a user-friendly graph querying system. In: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pp. 893–896. ACM (2014)
24. Yu, T., et al.: Spider: a large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. arXiv preprint [arXiv:1809.08887](https://arxiv.org/abs/1809.08887) (2018)
25. Zhong, V., Xiong, C., Socher, R.: Seq2sql: generating structured queries from natural language using reinforcement learning. arXiv preprint [arXiv:1709.00103](https://arxiv.org/abs/1709.00103) (2017)