SchenQL: A query language for bibliographic data with aggregations and domain-specific functions

Christin Katharina Kreutz
ckreutz@acm.org
Trier University
Trier, Germany

Martin Blum
Trier University
Trier, Germany

Ralf Schenkel
schenkel@uni-trier.de
Trier University
Trier, Germany

ABSTRACT
Current search interfaces of digital libraries are not suitable to satisfy complex or convoluted information needs directly, when it comes to cases such as “Find authors who only recently started working on a topic”. They might offer possibilities to obtain this information only by requiring vast user interaction.

We present SchenQL, a web interface of a domain-specific query language on bibliographic metadata, which offers information search and exploration by query formulation and navigation in the system. Our system focuses on supporting aggregation of data and providing specialized domain dependent functions while being suitable for domain experts as well as casual users of digital libraries.

CCS CONCEPTS
• Information systems → Query languages; Digital libraries, query formulation

KEYWORDS
domain-specific query language, digital libraries, query formulation

ACM Reference Format:
Christin Katharina Kreutz, Martin Blum, and Ralf Schenkel. 2022. SchenQL: A query language for bibliographic data with aggregations and domain-specific functions. In The ACM/IEEE Joint Conference on Digital Libraries in 2022 (JCDL ’22), June 20–24, 2022, Cologne, Germany. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3529372.3533282

1 INTRODUCTION
Nowadays, there is a plethora of digital libraries that provide bibliographic metadata such as the ACM DL1, Bibsonomy [7], dblp [10], Google Scholar2, Semantic Scholar3, SpringerLink4 or ResearchGate5. These interfaces support search, exploration and sometimes also simple query formulation beyond keyword-based search, but they fail to easily satisfy more complex information needs, e.g. “Find the papers written by any authors of two different institutions A and B” or “Find authors who only recently started working on a topic”.

A way to overcome these limitations of search interfaces of digital libraries is usage of specialised tools such as GrapAL [2], which allows query formulation via the graph-query language Cypher [4]. The big disadvantage with GrapAL is the prerequisite of all users having to be familiar with that all-purpose query language.

To help users access bibliographic metadata more easily, we extend our domain-specific query language and user interface SchenQL [9]. It focuses on aggregation of data and bibliographic functions, such as RELATED KEYWORDS TO ”DLs” or PUBLICATIONS WITH HIGHEST CORERANK METRIC. SchenQL offers simple language components which are easily learned by domain-experts as well as casual users of digital libraries. The queries resemble natural language and the graphical user interface helps in query construction. The two aforementioned information needs could be expressed by the following queries: PUBLICATIONS WRITTEN BY ANY DISTINCT 2 OF [(PERSON WORKS FOR ”University of Pisa”), (PERSON WORKS FOR ”National Research Council, Italy”)] and PERSONS AUTHORED (PUBLICATIONS ABOUT KEYWORD ”DLs”) AND AUTHORED NO (PUBLICATIONS ABOUT KEYWORD ”DLs” WITH YEAR AT MOST 2019).

2 SCHENQL
SchenQL [9] describes both the underlying domain-specific query language and the graphical user interface. We extend its previously presented functional range. Our main goal is to support all types of users of digital libraries in their information needs, even domain-experts who tend to use more sophisticated search options compared to casual users [15]. Thus, we offer a broad selection of domain-specific functionalities. Our system supports query formulation as well as information exploration but focuses more on the formulation as users of digital libraries have been found to prefer searching to browsing in some domains [15]. While our query language can be applied to any digital library, it has been designed with the dblp computer science bibliography7 as central use case. Therefore, some language components model the particularities of dblp, such as using keys for referencing publications and persons (e.g., “homepages/t/EdwardFox”) and using numerical suffixes for disambiguating author names (e.g., “Wei Wang 0042”).

2.1 Grammar
We extend and redefine SchenQL [9], but all previously described functionality remains (a hidden) part of it. On top of that, we add

1https://dl.acm.org/
2https://scholar.google.com/
3https://www.semanticscholar.org/
4https://link.springer.com/
5https://www.researchgate.net

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

JCDL ’22, June 20–24, 2022, Cologne, Germany
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9345-4/22/06... $15.00
https://doi.org/10.1145/3529372.3533282

6As of 2022, the GrapAL web interface is no longer supported.
7https://dblp.org/
Table 1: SchenQL grammar. Bold terms represent starting points for new rules, orange terms represent fixed language components, italic terms are variables of the type they describe. Black brackets, pipe symbols, plus symbols and question marks are only used for readability of the grammar. Brackets indicate scopes which signs refer to. Pipes indicate a choice between language components. Plus symbols indicate a comma-separated list of the preceding component. Question marks indicate that the preceding language component is optional.

| query | description |
|-------|-------------|
| **C** | conference anchor |
| **C.L** | conference literals |
| **C.F** | conference filters |
| **F** | full-text search for keywords |
| **K** | keyword anchor |
| **K.L** | keyword literals |
| **K.F** | keyword filters |
| **PU** | publication anchor |
| **PU.O** | publication object literals |
| **PU.F** | publication filters |
| **PE** | person anchor |
| **PE.C** | person object literals |
| **PE.E** | person entity literals |
| **PE.F** | person filters |
| **I** | institution anchor |
| **I.O** | institution object literals |
| **I.F** | institution filters |
| **M** | metric anchor |
| **M.A** | metric literals |
| **M.OP** | metric operator literals |
| **M.O** | metric operators |
| **M.COM** | metric comparative literals |

When observing keywords or words, we offer full-text search for words over titles and abstract (e.g. PUBLICATIONS ABOUT TERMS "(digital:libraries)") while keywords are a list of topics associated with the publication (e.g. PUBLICATIONS ABOUT KEYWORDS "(digital:library", "disq")

Filters on base concepts can be concatenated with Boolean operators (e.g. CONFERENCE WITH ACRONYM "JCDL" OR WITH YEAR 2000 or INSTITUTIONS WITH COUNTRY "DE" AND NOT WITH CITY "Cologne"). An AND operator is implicitly assumed between multiple filters and can be dropped (e.g. ARTICLES WITH YEAR AT LEAST 2010 AND WITH YEAR AT MOST 2020). 

Literals can be utilised to identify single or multiple entities of a specific base type. They can be used in filters instead of the longer formulation mentioning the base concept specifically (e.g. shortening PUBLICATIONS APPEARED IN (CONFERENCE WITH ACRONYM "JCDL") to PUBLICATIONS APPEARED IN "JCDL" by using the acronym literal). 

To enable different types of searches based on string literals like author names, certain prefixes can be used: ~ indicates a search, where all blank-separated specific query components need to be part of the name in any order (e.g. PERSONS NAMED ~"wang we1") returns persons with names Wei-Bo Wang and Wang Wei Lee, while incorporation of = before person names indicates a required exact match.
match of the name. If no prefix is used for search of person names, names with and without the numerical suffix are retrieved (e.g. PERSONS NAMED ="Wei Wang" returns the one person with the unique name Wei Wang but does not return Wei Wang 0042 while PERSONS NAMED "Wei Wang" returns both of them).

**Restriction** of the number of returned entities of queries can be achieved by either using LIMIT or ~RANK in specific positions in the grammar (see Table 1). The first variant specifies an upper limit for the number of results to return (e.g. 5 ARTICLES CITED BY “journals/interactions/Myers98”), the second variant observes ranks of resulting entities such that more than the specified amount of results can be returned if multiple occupy the same rank (e.g. CONFERENCES WITH ~5 HIGHEST H-AVG METRIC, which may return more than five results if some top conferences have the same H-AVG metric). Here, an application of the rank restriction is only possible if something is aggregated either by using an aggregation filter or the respective anchor points for keywords, publications, persons or institutions (via e.g. MOST CITED or MOST RESEARCHING).

SchenQL offers a number of **general functions** on bibliographic metadata that do not return entities of base concepts but other data: COUNT returns the number of entities in the following query, CORE RANKS FOR lists the core ranks and the number of occurrences for the following entities, ALTERNATIVE NAMES FOR returns the list of names for the following entities, MOST FREQUENT attribute OF returns the most frequent value of the specified attribute in the following query and attribute OF returns the values for the attributes of the following query.

The complete overview of SchenQL’s grammar with filters and query components can be found in Table 1. Values of non-numeric variables always need to be encompassed by quotation marks.

### 2.2 GUI

The SchenQL GUI is an updated form of the SchenQL GUI [9]. The search interface supports query construction with a suggestion and auto-completion feature (shown in Figure 1). Possible components to continue queries at any given point are suggested below, they can be clicked to appear as part of queries. In the component suggestions, we did not include the specialisations as this would overload the interface. Possible following language components are colour-coded. The colour indicates the type of the component e.g. base concepts (such as CONFERENCES or PERSONS), filters (such as WITH YEAR or ABOUT), literals (such as NUMBER or "STRING"), or restrictions (such as LIMIT or ~RANK). After running a SchenQL query, a search result is shown. Clicking on result items takes a user to the detail view of the specific type, Figure 2 shows a detail view of a person. Here we include details on the publications, citation information and keywords of their work. Additionally, the Ego Graph [12] visualisation (in the right part of Figure 2) highlights the most frequent co-authors of a person in the middle. The nodes surrounding the middle symbolise the co-authors, the shorter the distance between the middle and the co-author, the more works have been published together. Another visualisation included in person’s, publication’s, conference’s and journal’s detail views is the BowTie [8]. It can be used to estimate an entity’s position in the scientific landscape, as it visualises the number and age distribution of references and citations (see Figure 3 for a visual explanation).

### 2.3 System Architecture and Data

The SchenQL front-end uses Node.js 16.13 and incorporates Boot and Spring Boot. It communicates with a REST API which accepts user input queries and relays them to the compiler module. Here, for the lexer and parser, we extended and modified the previous SchenQL grammar which was generated with ANTLR to also support aggregation functions as well as more filters in SchenQL.
Table 2: Common (upper part) and specialised (lower part) information needs of users of digital libraries and SchenQL queries.

| Information need | SchenQL query formulation |
|------------------|---------------------------|
| Find person with 5 longest names | PERSON WITH 5 LONGEST NAME |
| Find authors of 5 earliest papers on topic | PERSONS WITHIN (~5 OLDEST (PUBLICATIONS ABOUT KEYWORD 'digital library')) |
| Find authors who only recently started working on topic | PERSONS AUTHORED (PUBLICATIONS ABOUT KEYWORD 'digital libraries') AND AUTHORED NO (PUBLICATIONS ABOUT KEYWORD 'digital libraries' WITH YEAR AT MOST 2017) |
| Find authors who published the most in the last three years | MOST RESEARCHING (PERSONS AUTHORED (PUBLICATIONS WITH YEAR AT LEAST 2015)) ABOUT KEYWORD 'digital library' |
| Find author co-citation | ARTICLES WRITTEN BY Walid Ammar APPEARED IN ‘JCDL’ |
| Find authors who cited a specific paper | ARTICLES (PUBLICATIONS CITED BY (PUBLICATIONS WRITTEN BY “Norbert Fuhr”)) REFERENCES (PUBLICATIONS WRITTEN BY “Norbert Fuhr”)) |
| Find persons with 5 longest names | PERSONS WITH 5 LONGEST NAME |
| Find papers on year X | PUBLICATIONS WITH YEAR X |
| Find papers with the highest h-index | INSTITUTIONS WITH MEMBERS (PERSONS AUTHORED (CONF/ICDM/ZHU2018)) |
| Find papers with the most citalion on topic | PUBLICATIONS ABOUT KEYWORD 'digital library' REFERENCES (PUBLICATIONS WITH COMMON KEYWORD 'digital library') |
| Find publications in conference in year | RELATIONAL KEYWORDS (PUBLICATIONS WITH YEAR X APPEARED IN ‘JCDL’) |
| Find publications in journal | RELATED KEYWORDS TO 'digital libraries' |

2.4 Application and Information Needs

SchenQL can be used to satisfy typical information needs of users of digital libraries or more complex ones which cannot directly be answered by current digital libraries.

Table 2 contains a selection of typical information needs (upper part) and ones which cannot be directly satisfied by using digital libraries (lower part) as well as their exemplary formulations in SchenQL. The listed information needs are partially mentioned in previous works [2, 3, 5, 17]. Further information needs such as query refinement, backwards and forwards citation analysis, quality judgement [6] as well as support in query formulation and general information exploration are mainly supported by the GUI of SchenQL.

3 CONCLUSION

We presented an extension of the SchenQL query language and GUI that incorporates aggregation functions and supports more general functions. Our system aims to support casual users as well as domain-experts by providing a huge selection of domain-specific functionalities. We make these functions accessible especially for casual users by suggesting possible following query components in our GUI.

SchenQL could be extended by enabling the support of graph-based operations (as seen with GrapAL [2]) such as the calculation of centrality scores, PageRank or hubs and authorities on a user-defined portion of the bibliographic network. Another line of work could focus on the incorporation of different visualisations to support users’ needs, for example colour-coded topical distributions in publications to help find relevant papers or estimate an author’s fit as a reviewer for a manuscript.

Evaluations of SchenQL could include information search where users are asked to retrieve information for predefined queries such as the ones we presented in Table 2 or more open, task-based studies where user behaviour and preferences could be observed more in detail as seen in other evaluations [6]. For example sense-making studies where users are asked to find and collect information [11] such as the question which one of two researchers could better fit the defined portion of the bibliographic network. Another line of work could focus on the incorporation of different visualisations to support users’ needs, for example colour-coded topical distributions in publications to help find relevant papers or estimate an author’s fit as a reviewer for a manuscript.
REFERENCES

[1] Waheed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hou-Han Ooi, Matthew E. Peters, Joanna Power, Sam Skjongberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the Literature Graph in Semantic Scholar. In NAACL-HLT (3). Association for Computational Linguistics, 84–91.

[2] Christine Bettis, Joanna Power, and Waheed Ammar. 2019. GraphAL: Connecting the Dots in Scientific Literature. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations, Marta R. Costa-jussà and Enrique Alfonsa (Eds.). Association for Computational Linguistics, 147–152. https://doi.org/10.18653/v1/p19-3025

[3] Stephan Bloehdorn, Philipp Cimiano, Alistair Duke, Peter Haase, Jörg Heizmann, Ian Thurlow, and Johanna Völker. 2007. Ontology-Based Question Answering for Digital Libraries. In Research and Advanced Technology for Digital Libraries, 11th European Conference, ECDL 2007, Budapest, Hungary, September 16-21, 2007, Proceedings (Lecture Notes in Computer Science, Vol. 4675), László Kovács, Norbert Fuhr, and Carlo Meghini (Eds.). Springer, 1433–1445. https://doi.org/10.1007/978-3-540-74851-9_2

[4] Nadine Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindauker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andres Taylor. 2018. Cypher: An Evolving Query Language for Property Graphs. In Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018, Gautam Das, Christopher M. Jermaine, and Philip A. Bernstein (Eds.). ACM, 1433–1445. https://doi.org/10.1145/3183713.3190657

[5] Sergio Gómez-Villamor, Gerard Soldevila-Miranda, Alexi Giménez-Vañó, Norbert Martínez-Bazan, Victor Muntés-Muñoz, and Josep Lluís Larriba-Pey. 2008. BIBEX: a bibliographic exploration tool based on the DEX graph query engine. In EDBT 2008, 11th International Conference on Extending Database Technology, Nantes, France, March 25-29, 2008, Proceedings (ACM International Conference Proceeding Series, Vol. 261), Alfonso Kemper, Patrick Valduriez, Noureddine Mouaddib, Jens Teubner, Mokrane Bouazghour, Volker Markl, Laurent Amsaleg, and Ioana Manolescu (Eds.). ACM, 735–739. https://doi.org/10.1145/1353348.1353439

[6] Orland Hoeker and Taraneh Khazaee. 2015. Evaluating citation visualization and exploration methods for supporting academic search tasks. Online Inf. Rev. 39, 2 (2015), 229–254. https://doi.org/10.1108/OIR-10-2014-0259

[7] Andreas Hotho, Robert Jäschke, Dominik Benz, Miranda Grafh, Beate Krause, Christoph Schmitz, and Gerd Stumme. 2009. Social Bookmarking am Beispiel BibSonomy. In Social Semantic Web: Web 2.0 - Was nun?. Andreas Blumauer and Tassilo Pellegrini (Eds.). Springer, 363–391. https://doi.org/10.1007/978-3-540-72216-8_18

[8] Taraneh Khazaee and Orland Hoeker. 2017. Supporting academic search tasks through citation visualization and exploration. Int. J. Digit. Libr. 18, 1 (2017), 59–72. https://doi.org/10.1007/s10799-016-0170-x

[9] Christin Katharina Kreutz, Michael Wölz, Jascha Knack, Benjamin Weyers, and Ralf Schenkel. 2021. SchenQL: in-depth analysis of a query language for bibliographic metadata. Int. J Digit Libr (2021). https://doi.org/10.1007/s10799-021-00317-8

[10] Michael Ley. 2009. DBLP - Some Lessons Learned. Proc. VLDB Endow. 2, 2 (2009), 1493–1500. https://doi.org/10.14778/1687553.1687577

[11] Peter Pirolli. 2009. Powers of 10: Modeling Complex Information-Seeking Systems at Multiple Scales. Computer 42, 3 (2009), 33–40. https://doi.org/10.1109/MC.2009.94

[12] Florian Reitz. 2010. A Framework for an Ego-centered and Time-aware Visualization of Relations in Arbitrary Data Repositories. CoRR abs/1009.5183 (2010). arXiv:1009.5183 http://arxiv.org/abs/1009.5183

[13] Arnab Sinha, Zihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Paul Hsu, and Kuansan Wang. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. In WWW (Companion Volume). ACM, 243–246.

[14] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. ArNetMiner: extraction and mining of academic social networks. In KDD. ACM, 990–998.

[15] Oksana Zavalina and Elena Vassilieva. 2014. Understanding the Information Needs of Large-Scale Digital Library Users. Library Resources & Technical Services 58 (04 2014), 84. https://doi.org/10.5860/IrLts2.84

[16] Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Bin Shao, Ru Li, and Kuansan Wang. 2019. OAG: Toward Linking Large-scale Heterogeneous Entity Graphs. In KDD. ACM, 2585–2595.

[17] Yongjun Zhu. 2017. Graph-based Interactive Bibliographic Information Retrieval Systems.