Semantische relaties verkennen in data op het web

Exploring Semantic Relationships in the Web of Data

Laurens De Vocht

Promotoren: prof. dr. ing. E. Mannens, dr. ir. R. Verborgh
Proefschrift ingediend tot het behalen van de graad van
Doctor in de ingenieurswetenschappen: computerwetenschappen

Vakgroep Elektronica en Informatiesystemen
Voorzitter: prof. dr. ir. R. Van de Walle
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2016 - 2017

UNIVERSITEIT
GENT

ISBN 978-94-6355-024-6
NUR 988, 995
Wettelijk depot: D/2017/10.500/59

Members of the examination committee

Chair

prof. dr. ir. Patrick De Baets (Ghent University)

Reading Committee

prof. dr. Peter Lambert (Ghent University, *secretary*)
dr. ir. Toon De Pessemier (Ghent University)
dr. ir. Bert Van Nuffelen (Tenforce)
prof. dr. ir. Martin Ebner (Graz University of Technology)

Other members

prof. dr. ing. Erik Mannens (Ghent University, *promotor*)
dr. ir. Ruben Verborgh (Ghent University, *promotor*)

Preface

Ever since I decided to specialize in computer science engineering, I have been intrigued by how people interact with computers and machines. Around the same time, the first smartphones with only a touchscreen were released and social networks like Twitter and Facebook were breaking through. During my Masters, I learned that the user experience in applications on a smartphone, or any other device, is often driven by how data is integrated in the front-end and back-end, and how data is structured 'behind the scenes'. In my Master thesis, I investigated how data from social media could be used to add a dynamic, real-time context during scientific conferences to the more static data contained in digital libraries. The focus lied on connecting researchers with potentially relevant publications. Key to the approach was the use of semantic annotations to interlink data sources.

Interlinking data sources with semantics is not only useful for the user experience but also allows more interoperability between machines. Using the same semantics between users and machines guarantees (real-world) things are being referred to uniquely and interpreted correctly. Working with semantic annotations introduces new challenges and opportunities when it comes to interacting and exploring data. This is the subject of this PhD.

About six years ago, moments before I defended my Master thesis, my supervisor at that time, prof. Erik Duval, asked me: "A PhD, wouldn't you consider it?". It wasn't until I met prof. Erik Mannens, I was finally convinced to 'go for it'. Prof. Erik Mannens' presence alone is motivating and his enthusiasm about the work we do in the lab is extraordinary. My co-supervisor dr. Ruben Verborgh's eye for detail and accurate feedback was extremely important, not only for this dissertation but for almost all publications we worked on. The opportunity to pursue a PhD prof. Erik Mannens offered me, allowed me to continue the research I started at TUGraz during my Master thesis with Selver Softic.

The research with Selver resulted in a tool for exploring research. The tool focuses on a use case applying the Web 2.0 to scientific research. We looked into how well dynamic data sources, like social media or data of "call for papers" can be brought in line with more static data sources like metadata from academic libraries. The data source "Conference Linked Data" (COLINDA) that Selver developed, with information on the calls for papers of many scientific conferences, proved to be very useful for interlinking this data.

With dr. Christian Beecks from RWTH Aachen, I worked on investigating the role of heuristics in path-based storytelling, the optimization of link estimation between facts in a path-based story by increasing the consistency of links between facts.

From the beginning, I had the pleasure to work with Raf Buyle, who showed me the intricacies on how to get the right people to work together for research on semantics to be picked-up and embedded into governance. In particular the work we did for "Open Standards for Linked Organizations" (OSLO), aimed to capture the fundamental characteristics of information exchange for public administrations on all levels. OSLO resulted in vocabularies and application profiles for the exchange of information about people, addresses, organizations and public services. Together with Mathias Van Compernolle and dr. Peter Mechant from MICT we worked on several related iMinds and imec projects that put the research on linked data in a more practical and organizational perspective, especially for public administrations in Flanders.

Thanks to all my colleagues whom I worked with during the past five years: Davy, Sam, Hajar, Miel, Tom, Anastasia, Pieter C., Pieter H., Dieter D.W., Dieter D.P., Ruben T., Ben, Joachim, Doërthe, Martin, Gerald, Sven, Julian and everyone else from IDLab in the other offices. In particular Laura, Ellen and Kristof for supporting us with our administration and IT practicalities.

To all members of the jury: prof. Patrick De Baets, prof. Peter Lambert, dr. Toon De Pessemier, dr. Bert Van Nuffelen, and prof. Martin Ebner, I would like to express my gratitude for thoroughly reviewing the work and results presented in this dissertation. Based on your feedback and questions, I made additional changes and clarifications to several parts of this book.

Thanks to all the people who worked with me on one or more conference contributions or other papers, and everyone who I met at conferences or other events where we discussed linked data, its exploration and so many other subjects.

Thanks to my friends for sticking around and being sincerely interested in this PhD that occupied me, but most importantly for regularly distracting me from it.

I would like to thank my family for their support and endless patience, in particular during the times when I was physically present but my thoughts were not.
My grandparents who always believed in me, but were yet each time extremely impressed and always so proud on the things I have been doing.
My parents Erik and Véronique, sisters Benedikte and Judith, and my brother Ludo.
Last but not least, my partner Laurence, for being there to support me along the entire way, ever since we met.

Laurens De Vocht

June 2017,
Gent, Belgium.

Contents

List of Figures

List of Tables

List of Acronyms

AP	Average (Search) Precision.
API	Application Programming Interface.
BGP	Basic Graph Pattern.
CSV	Comma Separated Values.
DBO	DBpedia Ontology.
DBP	DBpedia Resource.
DC	Dublin Core.
EDA	Exploratory Data Analysis.
FOAF	The Friend Of A Friend ontology.
HTML	HyperText Markup Language.
HTTP	HyperText Transfer Protocol.
HYP	Hypothesis.
IR	Information Retrieval.
IRI	Internationalized Resource Identifier, a generalization of URI.
JSON	JavaScript Object Notation.
JSON-LD	JSON for Linked Data.
KV	Key variable.
LD	Linked Data.
LOD	Linked Open Data.
MAP	Mean Average (Search) Precision.
MUTO	Modular Unified Tag Ontogy.
N3	Notation3, an RDF serialization.
NDD	Normalize DBpedia Distance.
NFD	Normalized Freebase Distance.
NGD	Normalized Google Distance.

NSWD	Normalized Semantic Web Distance.
OWL	Web Ontology Language.
QA	Question Answering.
RDF	Resource Description Framework.
RDFA	RDF Annotations – a notation used to embed RDF in (X)HTML pages.
RDFS	RDF Schema.
RQ	Research Questions.
SIOC	Semantically Interlinked Online Communities.
SPARQL	SPARQL Protocol and RDF Query Language.
SQL	Structured Query Language.
SWRC	Semantic Web for Resource Communites.
TF-IDF	Term Frequency - Inverse Document Frequency.
TURTLE	Terse RDF Triple Language, an RDF serialization.
URI	Universal Resource Identifier.
W3C	World Wide Web Consortium.
XHTML	Extensible HyperText Markup Language.
XML	Extensible Markup Language.

Summary

After the launch of the World Wide Web, it became clear that searching documents on the Web would not be trivial. Well-known engines to search the web, like Google, focus on search in web documents using keywords. The documents are structured and indexed to ensure keywords match documents as accurately as possible. However, searching by keywords does not always suffice. It is often the case that users do not know exactly how to formulate the search query or which keywords guarantee retrieving the most relevant documents. Besides that, it occurs that users rather want to browse information than looking up something specific. It turned out that there is need for systems that enable more interactivity and facilitate the gradual refinement of search queries to explore the Web. Users expect more from the Web because the short keyword-based queries they pose during search, do not suffice for all cases.

On top of that, the Web is changing structurally. The Web comprises, apart from a collection of documents, more and more linked data, pieces of information structured so they can be processed by machines. The consequently applied semantics allow users to exactly indicate machines their search intentions. This is made possible by describing data following controlled vocabularies, concept lists composed by experts, published uniquely identifiable on the Web. Even so, it is still not trivial to explore data on the Web. There is a large variety of vocabularies and various data sources use different terms to identify the same concepts.

This PhD-thesis describes how to effectively explore linked data on the Web. The main focus is on scenarios where users want to discover relationships between resources rather than finding out more about something specific. Searching for a specific document or piece of information fits in the theoretical framework of information retrieval and is associated with exploratory search. Exploratory search goes beyond 'looking up something' when users are seeking more detailed understanding, further investigation or navigation of the initial search results. The ideas behind exploratory search and querying linked data merge when it comes to the way knowledge is represented and indexed by machines – how data is structured and stored for optimal searchability. Queries and information should be aligned to facilitate that searches also reveal connections between results. This implies that they take into account the same semantic entities, relevant at that moment. To realize this, we research three techniques that are evaluated one by one in an experimental set-up to assess how well they succeed in their goals. In the end, the techniques are applied to a practical use case that focuses on forming a bridge between the Web and the use of digital libraries in scientific research.

Our first technique focuses on the interactive visualization of search results. Linked data resources can be brought in relation with each other at will. This leads to complex and diverse graphs structures. Our technique facilitates navigation and supports a workflow starting from a broad overview on the data and allows narrowing down until the desired level of detail to then broaden again. To validate the flow, two visualizations where implemented and presented to test-users. The users judged the usability of the visualizations, how the visualizations fit in the workflow and to which degree their features seemed useful for the exploration of linked data.

There is a difference in the way users interact with resources, visually or textually, and how resources are represented for machines to be processed by algorithms. This difference complicates bridging the users' intents and machine executable queries. It is important to implement this 'translation' mechanism to impact the search as favorable as possible in terms of performance, complexity and accuracy. To do this, we explain a second technique, that supports such a bridging component. Our second technique is developed around three features that support the search process: looking up, relating and ranking resources. The main goal is to ensure that resources in the results are as precise and relevant as possible. During the evaluation of this technique, we did not only look at the precision of the search results but also investigated how the effectiveness of the search evolved while the user executed certain actions sequentially.

When we speak about finding relationships between resources, it is necessary to dive deeper in the structure. The graph structure of linked data where the semantics give meaning to the relationships between resources enable the execution of pathfinding algorithms. The assigned weights and heuristics are base components

of such algorithms and ultimately define (the order) which resources are included in a path. These paths explain indirect connections between resources. Our third technique proposes an algorithm that optimizes the choice of resources in terms of serendipity. Some optimizations guard the consistence of candidate-paths where the coherence of consecutive connections is maximized to avoid trivial and too arbitrary paths. The implementation uses the A* algorithm, the de-facto reference when it comes to heuristically optimized minimal cost paths. The effectiveness of paths was measured based on common automatic metrics and surveys where the users could indicate their preference for paths, generated each time in a different way.

Finally, all our techniques are applied to a use case about publications in digital libraries where they are aligned with information about scientific conferences and researchers. The application to this use case is a practical example because the different aspects of exploratory search come together. In fact, the techniques also evolved from the experiences when implementing the use case. Practical details about the semantic model are explained and the implementation of the search system is clarified module by module. The evaluation positions the result, a prototype of a tool to explore scientific publications, researchers and conferences next to some important alternatives.

Samenvatting

Na de lancering van het wereldwijde Web werd het duidelijk dat zoeken naar documenten op het Web geen evidentie zou zijn. Met alombekende zoekmachines voor het Web, zoals Google, kunnen gebruikers met sleutelwoorden zoeken in webdocumenten. Daarvoor worden de documenten zodanig gestructureerd en geïndexeerd dat de sleutelwoorden er zo nauwkeurig mogelijke raakpunten mee kunnen hebben. Het zoeken op sleutelwoorden volstaat echter niet altijd. Vaak is het zo dat gebruikers niet exact weten hoe ze een zoekopdracht best formuleren of welke sleutelwoorden ze nodig hebben om relevante documenten kunnen terugvinden. Daarnaast gebeurt het ook dat gebruikers eerder willen bladeren door informatie dan iets specifiek opzoeken. Er waren systemen nodig zijn die meer interactiviteit creëren en het mogelijk maken om geleidelijk aan zoekopdrachten te verfijnen om het Web te kunnen verkennen. Gebruikers verwachten meer van het Web omdat de korte sleutelwoord-gebaseerde vragen die ze kunnen stellen tijdens het zoeken, niet steeds volstaan.

Bovendien is het Web structureel aan het veranderen. Naast een verzameling van documenten, bestaat het Web steeds meer uit gelinkte data (*linked data*), stukjes informatie die zodanig gestructureerd en beschreven zijn dat ze door machines kunnen verwerkt worden. De daarvoor toegepaste semantiek laat gebruikers toe om exact aan machines aan te geven waarnaar ze op zoek zijn. Dit gebeurt door data te beschrijven met gecontroleerde vocabularia, door experts samengestelde lijsten van vastgelegde concepten, uniek identificeerbaar op het Web gepubliceerd. Toch is het daarmee nog niet evident om data via het Web te verkennen. Er is immers een ruime verscheidenheid aan vocabularia en bronnen gebruiken regelmatig verschillende termen om dezelfde concepten te benoemen.

Deze doctoraatsthesis beschrijft hoe het verkennen van het Web van gelinkte data te realiseren op een effectieve manier. De voornaamste focus ligt op scenario's waar

gebruikers verbanden tussen zaken (*resources*) willen ontdekken, eerder dan meer informatie te weten komen over één iets specifiek. Het zoeken naar één specifiek document of stukje informatie past in het typische theoretische kader van het 'ophalen van informatie' (*information retrieval*) en sluit aan bij verkennend zoeken. Verkennend zoeken gaat verder dan louter 'iets opzoeken' wanneer gebruikers een diepgaander begrip, verder onderzoek of navigatie van de initiële zoekresultaten vereisen. De ideeën achter verkennend zoeken en het bevragen van gelinkte data komen samen wanneer het gaat over hoe kennis wordt voorgesteld en door machines wordt geïndexeerd – of hoe gegevens gestructureerd en bijgehouden worden met het oog op optimale doorzoekbaarheid. Om mogelijk te maken dat zoekopdrachten en vragen die gebruikers stellen ook verbanden tussen resultaten onthullen, moeten zowel de vraagstelling (*query*) als de informatie op elkaar afgestemd worden. Dit betekent dat ze rekening houden met dezelfde semantische entiteiten, die op dat moment relevant zijn. Om dit te realiseren, worden er drie technieken onderzocht en één voor één geëvalueerd in een experimentele opstelling om te valideren of ze slagen in hun opzet. Deze technieken worden uiteindelijk toegepast in een praktische use case waar een brug wordt geslagen tussen het Web en het gebruik van digitale bibliotheken in wetenschappelijk onderzoek.

Onze techniek die het eerst besproken wordt gaat over het interactief visualiseren van zoekresultaten. Bij gelinkte data, kunnen zaken naar believen met elkaar in relatie worden gebracht. Dit leidt vaak tot complexe, gevarieerde graafstructuren. Om gebruikers te helpen bij het navigeren, ondersteunt de techniek een workflow die vertrekt van een ruim overzicht op de data en laat toe om de scope te vernauwen tot op het gewenste detailniveau en vervolgens terug te verbreden. Om de flow te valideren werden twee visualisaties geïmplementeerd en voorgesteld aan de gebruikers. De gebruikers beoordeelden de bruikbaarheid van de visualisaties, maar ook hoe de visualisaties pasten in de workflow en in welke mate hun functionaliteiten nuttig bleken bij het verkennen van gelinkte data.

Er is een groot verschil tussen hoe gebruikers zaken te zien krijgen – visueel of tekstueel – en hoe dezelfde zaken gerepresenteerd worden voor machines om door algoritmen verwerkt te kunnen worden. Dit verschil maakt het niet evident om de brug te slaan van de intenties van gebruikers naar vraagstellingen en acties die een machine kan uitvoeren. Het komt eropaan dit 'vertaalmechanisme' zodanig te implementeren dat de impact op het zoeken zo gunstig mogelijk is zowel wat betreft performantie, complexiteit als nauwkeurigheid. Om een dergelijke brugcomponent te ondersteunen, wordt een tweede techniek onderzocht. Onze tweede techniek is ontwikkeld rond drie functionaliteiten die het zoekproces ondersteunen: opzoeken -, in verband brengen - en rangschikken van zaken tijdens het zoeken. Het voornaamste doel is ervoor zorgen dat de zaken die in de resultaten voorgesteld worden aan de gebruiker zo relevant en precies mogelijk zijn. Tijdens de evaluatie van deze techniek werd er niet alleen gekeken naar de kwaliteit van de zoekresultaten, maar ook hoe de effectiviteit van het zoeken evolueerde terwijl bepaalde gebruiker-acties stapsgewijs werden uitgevoerd.

Zodra het gaat om het vinden van verbanden tussen zaken, is het nodig om verder in de structuur te duiken. De graafstructuur van gelinkte data waarbij de semantiek betekenis geeft aan de relaties tussen zaken maakt het mogelijk om pad-algoritmes los te laten. De gewichten en heuristieken die worden toegekend en die basiscomponenten zijn van dergelijke algoritmes bepalen uiteindelijk welke zaken al-dan niet in de paden worden opgenomen. Deze paden verklaren onrechtstreekse verbanden tussen zaken. Onze derde techniek stelt een basisalgoritme voor die de keuze van zaken optimaliseert in functie van hun 'toevalstreffer'-gehalte (*serendipity*). Enkele optimalisaties waken over de consistentie van kandidaat-paden waarbij over de samenhang van de opeenvolgende verbanden wordt gewaakt om te triviale en te willekeurige paden te vermijden. De implementatie maakt gebruik van het A*-algoritme, de de-facto referentie als het gaat over heuristisch geoptimaliseerde minimale-kost paden. De effectiviteit van de paden werd gemeten op basis van een aantal gangbare automatische metrieken en aan de hand van surveys waarbij gebruikers de voorkeur konden aangeven voor paden, die telkens op een andere manier gegenereerd werden.

Ten slotte worden onze technieken toegepast in een use case omtrent publicaties in digitale bibliotheken, waar ze met informatie over wetenschappelijke conferenties en onderzoekers in verband worden gebracht. De toepassing in deze use case is een praktisch voorbeeld omdat hier de verschillende aspecten van verkennend zoeken samenkomen. Verder zorgden de ervaringen met het uitwerken van de use case ook voor de verdere ontwikkeling van de technieken. Praktische details omtrent het semantisch model worden uiteengezet en de implementatie van het zoeksysteem wordt module per module uitgelegd. De evaluatie plaatst het resultaat, een prototype van een hulpmiddel om wetenschappelijke publicaties, onderzoekers en conferenties te verkennen, naast een aantal belangrijke alternatieven.

Chapter 1

Introduction

I start in the middle of a sentence and
move both directions at once.

—John Coltrane.

The Web as a huge collection of linked documents [2] created new challenges for information retrieval. To address these challenges, Brin and Page introduced Google in 1998, the most popular large-scale Web search engine. Google heavily uses the links present in 'hypertext' documents and is designed to automatically crawl and index the Web efficiently [4].

In many cases when users search for information, it is very hard to exactly pinpoint (the document containing) the specific pieces of information they are looking for. In other cases users rather try to explore information than looking for a specific piece of information. Hence, users cannot realistically construct their intended search query correctly at the first attempt. "Users demand more of Web services, short queries typed into search boxes are not robust enough to meet all of their demands" [12]. Studies of early hypertext systems distinguished various search strategies and argued that "defining a hybrid system that guides discovery seems an appropriate compromise, but involves a number of trade-off decisions; how deeply the database is indexed, if some automatic controlled vocabulary is included, and how feedback is summarized and even formatted on the screen affect the strategies users will apply" [13]. Typically, when users formulate search queries to find relevant content on the Web, they intend to address a single target source that needs to match their entire query. In cases when users want to discover and explore resources across the Web they often need to repeat many sequences of search, check and rephrase until they have precisely refined their searches. In short, users need a system which facilitates iteratively refining what they are searching for.

Furthermore, the Web is changing. The Web is no longer only a huge collection of documents [1], but also more and more linked data [3], a 'Web of Data'. Linked data may be represented specifically for machine processing using the "Resource Description Framework" (RDF) [11] besides human-friendly readable representations in web documents. Linking data instead of documents introduces nevertheless (i) additional complexity when searching for information; and (ii) enables distributing search tasks across datasets directly benefiting from a semantic description.

Therefore, this PhD proposes a set of complementary techniques, each addressing a 'layer' of the search: (i) focusing on the user interface; (ii) acting as an intermediary; or (iii) taking care of the actual retrieval. We explain how each technique supports Web applications in fulfilling exploratory searches effectively. This is validated by measuring the user effectiveness, precision of search results, and the impact of certain features when isolated in terms of performance. The general focus is the iterative exploration of linked data spread across different data sources on the web. The generic methods and techniques developed in this PhD thesis find an application in various areas, among others in scientific research, industry, and media. Some example applications in each of these sectors:

- **Scientific Research.** The evolution of the Web enabled a wide range of users via wikis, blogs and other content publishing platforms to become content providers. Research data and many publications are publicly available online, not only via institutional repositories. However, data is often being stored in separate silos, a so-called 'walled garden' of platforms and institutional repositories for 'Science 2.0'. Combining information sources leads to mismatches between vocabularies and data structures of the different sources [10]. For example, a linked open data project for the Department of Economy, Science and Innovation Flanders (EWI), in which we participated, consisted of a use case on academic library, publications and research project metadata. The use case focused on the integration of a search and exploration interface for open data. Users had to interact with the data as graphs. The integrated workflow consisted of several aligned visualizations and facilitated dealing with such datasets. The project's outcome resulted in an article for the popular scientific Dutch EOS magazine in March 2015 [6].

- **Industry.** Data intensive applications, for example in the pharmaceutical-industry, involve many partners in the development of a product and benefit from embedding interactive and exploratory data visualizations in industry search applications. Typically this data is very well structured or has high quality meta-data about a variety of aspects, such as the clinical trials, compounds and processes. But it is complex to build systems that integrate and align this variety of data. We participated in a project named "Semantic Query Engine for Life Sciences" (SEQUEL), in cooperation with the company Ontoforce[1]. The goal was to gain deeper insight into query federation and the joining of (distributed) search results. The project investigated different back-end storage solutions [8, 9] for Ontoforce's semantic search platform DisQover[2]. DisQover is an exploratory search engine developed for the domain of life sciences and allows researchers to access and discover biomedical data. This leads to new insights in medicine and drug development [14]. Until the start of the project, companies mainly linked their privately held data into their own, often closed, semantic framework (if any at all). However, a multitude of relevant linked (open) data about drugs, chemicals and medical publications became available in the meantime. Particular attention went to mapping the technical requirements for implementing such a framework and developing a reusable and reproducible benchmark. This allowed adapting the query interface to the latest advances in database technology.

- **Media and Entertainment.** The media and entertainment sector can benefit from exploratory search techniques: when recombining data from multimedia archives or social media for storytelling, new hidden relations and trends among existing sources could be discovered. This enables application developers to design a whole range of interesting and entertaining applications and visualizations [15]. I participated in the project 'Towards a sustainable mobile tourism guide' [7]. The goal of the research project, consisting of a consortium of parties from Flanders, was twofold:

  (i) to stimulate innovation in the (mobile) tourism sector; and

[1] http://ontoforce.com

[2] http://disqover.com

(ii) to identify a sustainable solution for developing such innovations.

> Particular attention went to creating a reusable data model for mobile tourism guides to obtain data from many data sources. Many digital innovations have a recurring approach in regard to content production: digitize information relevant to the application at hand using some form of content management system and linking this digital content to a mobile application for example. The process of digitalizing the information and entering it into the content management system takes a considerable time investment. At the time of writing, most mobile applications have their own content management system built custom to the needs of the application itself, thus limiting the reusability for future applications as well as the reusability of the data inside.

Regardless of the application area, transitioning from traditional web search and retrieval to exploratory Semantic Web search is challenging [5]. More and more use cases and scenarios on the Web appear where exploratory search is beneficial. The additional required effort put into pre-processing, generating, interlinking and maintaining data sources as linked data, improves data re-usability and ensures that the methods and techniques for exploratory search are applicable to other domains. Each technique was refined in an applied and experimental setting where the set-up, data and queries were chosen carefully to address certain aspects of exploratory search (as shown in Figure 1.1): front-end (a), back-end (c), or as bridge in-between (b).

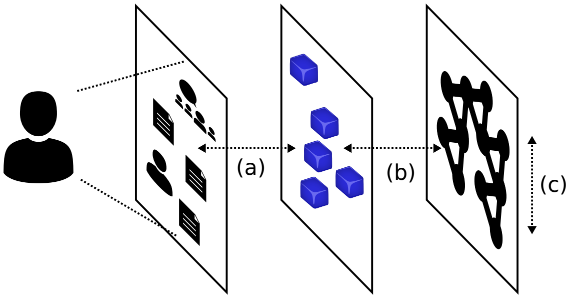


Figure 1.1: The techniques address a different level of the exploration: interactive visualization of search results in the front-end (a); processing queries to bridge front-end and back-end (b); and retrieving data and finding relationships through path-based storytelling in the back-end (c).

This PhD thesis consists of 4 parts and has 9 chapters in total:

- **Part I – Fundamentals** introduces the core concepts of exploratory search in Chapter 2 and explains the basics of linked data. The application of query techniques leads to the research questions. How linked data can be queried and the relation to exploratory search is outlined in Chapter 3.
- **Part II – Techniques** investigates three complementary query techniques, each contributing to exploratory search from a different perspective: user, machine, and in-between both. Each technique is presented following the same structure: (i) Introduction; (ii) Architectural Model: conceptual description and specification of the technique; (iii) Implementation: the practical application of the technique to the Web of data; and (iv) the Evaluation.

  Chapter 4 focuses on our technique for interactive search visualization and investigates how various exploration principles are perceived by users.

  Chapter 5 describes our proposed way to bridge the front-end and back-end of exploratory search.

  Chapter 6 studies revealing facts about how search results relate to each other. Strong emphasis lies on the path-based aspect of our technique, which is a possibility due to the graph structure of linked data.
- **Part III – Use Case** focuses on the application of exploratory search in academic libraries for scientific research. The use case itself is described in Chapter 7 and the implementation combining all three techniques is elaborated on in Chapter 8.
- **Part IV – Conclusions** in Chapter 9 reflects on our presented techniques, and explains how they provide answers to the research questions posed in Chapter 2.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[2] T. Berners-Lee. The world-wide web. *Computer Networks and {ISDN} Systems*, 25(4–5):454–459, 1992.

[3] C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[5] L. De Vocht. Iterative query refinement for exploratory search in distributed heterogeneous linked data. eng. *Doctoral Consortium at the 14th International Semantic Web Conference, Proceedings*, CEUR-WS, Bethlehem, PA, United States, 2015.

[6] L. De Vocht. Slim ecosysteem maakt vastgeroeste data los. dut. In. Volume 32 of EOS MAGAZINE 3, 2015, pages 104.

[7] L. De Vocht, R. Verborgh, E. Mannens, R. Van de Walle, W. Van den Bosch, R. Buyle, and B. Koninckx. Providing interchangeable open data to accelerate development of sustainable regional mobile tourist guides. *Proceedings of the 9th International Conference on Theory and Practice of Electronic Governance*. ICEGOV '15-16, pages 195–198, ACM, Montevideo, Uruguay, 2016.

[8] D. De Witte, L. De Vocht, K. Knecht, F. Pattyn, H. Constandt, E. Mannens, and R. Verborgh. Scaling out ETL queries for life science data in production. *Proceedings of the 9th Semantic Web Applications and Tools for Life Sciences International Conference*, Amsterdam, Netherlands, December 2016.

[9] D. De Witte, L. De Vocht, R. Verborgh, K. Knecht, F. Pattyn, H. Constandt, E. Mannens, and R. Van de Walle. Big linked data ETL benchmark on cloud commodity hardware. *Proceedings of the International Workshop on Semantic Big Data*. SBD '16, pages 12:1–12:6, ACM, San Francisco, California, 2016.

[10] D. M. Herzig and T. Tran. Heterogeneous web data search using relevance-based on the fly data integration. A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, and S. Staab, editors, *WWW*, pages 141–150, ACM, 2012.

[11] G. Klyne and J. J. Carrol. Resource Description Framework (RDF): concepts and abstract syntax. Recommendation. World Wide Web Consortium, February 2004. `http://www.w3.org/TR/rdf-concepts/`

[12] G. Marchionini. Exploratory search: from finding to understanding. *Commun. ACM*, 49(4):41–46, April 2006.

[13] G. Marchionini and B. Shneiderman. Finding facts vs. browsing knowledge in hypertext systems. *Computer*, 21(1):70–80, January 1988.

[14] F. Pattyn, S. Vandeschaeve, S. Vermaere, P. Van Huffel, K. Knecht, and H. Costandt. Semantic linking and integration of researchers and research organizations in disqover. *Proceedings of the 9th Semantic Web Applications and Tools for Life Sciences International Conference, Amsterdam NL, December 5-8, 2016.* 2016.

[15] M. Vander Sande, R. Verborgh, S. Coppens, T. De Nies, P. Debevere, L. De Vocht, P. De Potter, D. Van Deursen, E. Mannens, and R. Van de Walle. Everything is connected: using Linked Data for multimedia narration of connections between concepts. B. Glimm and D. Huynh, editors, *Proceedings of the 11th International Semantic Web Conference Posters and Demo Track*. Volume 914 of CEUR Workshop Proceedings, CEUR-WS.org, 2012.

Part I

Fundamentals

Chapter 2

Exploratory Search

> *Exploratory search makes us all pioneers and adventurers in a new world of information riches awaiting discovery along with new pitfalls and costs.*
>
> —Gary Marchionini.

This chapter starts with a background and theoretic conceptualization of exploratory search, giving a clear understanding of the term and continues with an overview on several applications of exploratory search to the Web in a variety of domains, each with their own focus. It appears that most practical implementations for the Web are isolated or at least in disarray. This leads to the identification of the research questions in this doctoral dissertation.

2.1 Background

Before diving into exploratory search and ways to implement it, we take a step back and look into the term 'Web search'. Systems developed to search for information on the Web, the so-called *'Web search engines'*, are popular and visible information retrieval applications. The goal of these engines can be seen as serving *lookup search tasks*, which correspond to a classic model of information retrieval. Figure 2.1 shows the conceptualization of this kind of retrieval according to Bates [3].

Each search process has a connection with the task that generates it [34]. The search effectiveness is associated with the 'relevance' of items in the search, which expresses how well a certain document matches an information need [35], in other

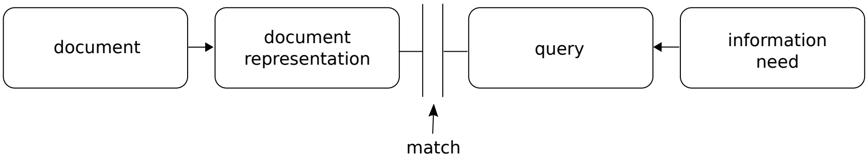


Figure 2.1: Bates' look-up based search model [3].

words, handling the search as an information retrieval process [29]. The *search effectiveness* is expressed as the "*degree of relevant retrieved documents in relation to the total number of retrieved documents (precision) and degree of relevant retrieved documents in relation to all the documents available (recall)*" [28].

Measuring recall and precision when all relevance judgments are available is possible, especially with a limited sets of documents. The provision of such judgments rapidly becomes impracticable when the size of the document set increases. This is particularly the case for the Web: back in 2005 the total number of indexable documents was estimated at 11.5 billion [12], in 2011 the largest estimated search index had peaks up to 50 billion web pages [5]. Simulation approaches could be used to measure performance of web access and search methods [27]. Hence, practically measuring recall is only possible relative to a sampled or pooled set of documents, not the Web as a whole [8].

Search tasks and activities are not limited to lookup, many real-life tasks contain multiple iterations, browsing results, finding relationships and detailed examination of results. This cannot be captured by Bates model on its own [38]. Marchionini introduced an extension of Bates' search model, incorporating the various different aspects of user-system interaction during exploratory search [20], a simplified version is presented in Figure 2.2. Marchionini identified *lookup* tasks such as known-item search, navigation, verification and question answering and associated them exploratory search in two dimensions: learn (comparison, integration, knowledge acquisition...) and investigate (analysis, accretion, synthesis...). In Part II, Techniques, we refer to methods and actions to support learning and investigation under the terms *relate* and *expand*. Each action corresponds to what Bates defined as "a move", each of them "a part of information searching" and "the basic unit of analysis of search behavior considered" [4]. Following this definition we consider "typing in a search term formulation" (for lookup) as an action.

An exploration session may start the search from a vague but still goal-oriented

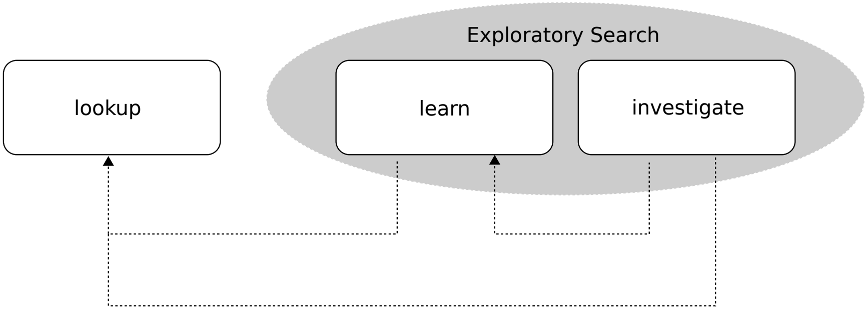


Figure 2.2: Associations between search activities according to Marchionini [20].

defined information need and users are able to refine their need upon the availability of new information to address it [9]. Task-oriented search scenarios go beyond retrieving information when a one-time perception of search tasks is neither possible nor sufficient. Such scenarios typically need further investigation, navigation or understanding of the search results. This requires that the data is presented first in an initial overview map that can be used as a starting point for further differentiation, learning and interpretation of the results to achieve a search task's goal.

Exploratory search represents "... a shift from the analytic approach of query-to-document matching to direct guidance at all stages of the information-seeking process." [37], where users can at all stages see immediate impact of their decisions. By following hyperlinks, users can better state and precise their information problem, and bring it closer to resolution. Exploratory search can describe either the context that motivates the search or the process by which the search is conducted [20].

Most work about exploratory and semantic search focuses explicitly more on the front-end or on the back-end. Typically they handle a single or few datasets and mainly concern preprocessing, structuring or indexing data. In exploratory and semantic search two aspects are equally important: representation of data for search and the way exploratory search actually takes place with the data (instead of documents). Table 2.1 lists related work on exploratory search with key variables, the core contribution and the datasets used to test the approach.

Table 2.1: Existing work on exploratory and semantic search has each it own distinct focus in terms of key variables, domain and main contribution. The column with test data sources lists the datasets that were used to evaluate the referenced system.

Reference	Year	Main Contribution	Test Data Sources	Key Variables
mSpace [30]	2005	Interactive Faceted Browsing for hypertext exploration with the aid of the Semantic Web.	Music Library Metadata	Interactivity
Sindice [26]	2008	indexing infrastructure with a Web frontend and API to locate SemanticWeb data sources such as RDF files and SPARQL endpoints	web crawler	Efficiency, Index Quality
Hermes [33]	2009	Translating keyword queries to structured queries based on an integrated schema of heterogeneous data sources	DBLP[a], Freebase[b], DBpedia[c], semanticweb.org [d]	Efficiency, Effectiveness
RelFinder [13]	2009	Systematic analysis of relationships in large knowledge bases	DBpedia	Interactivity
Waitelonis et al. [36]	2010	navigate and explore video data enriched with linked data along guided routes.	DBpedia, Yovisto[e] annotated videos	Effectiveness
SWSE [15]	2011	Distributed keyword - and focus query processing	web crawler	Efficiency
Li [17]	2012	Ranked top-k answers	semanticweb.org	Effectiveness
PowerAqua [18]	2012	Ontology-based question answering system, exploiting large, distibuted semantic web resources	DBpedia, 500+ distributed semantic documents[f]	Effectiveness
Discovery Hub [23]	2013	Faceted search drawing attention to initial query	DBpedia	Efficiency
Pinta [10]	2013	Uni-focal semantic browsing interface for exploratory search through several data sets linked via domain ontologies.	DBpedia, DBTune[g], Amazon Reviews	Interactivity, Effectiveness
LODMilla [24]	2013	Users can navigate and explore multiple LOD datasets and they can also save LOD views and share them with other users	DBpedia, DBLP, National Hungarian Data Archive[h].	Interactivity
SemFacet [1]	2016	Theoretical faceted search foundation in RDF, establish computational complexity, updating faceted interfaces: critical in the formulation of meaningful queries.	Yago[i]	Efficiency
Aemoo [25]	2017	Encyclopedic Knowledge Patterns (EKPs) as relevance criteria for selecting, organising, and visualising knowledge. EKPs are instantiated by mining Wikipedia following ontology patterns.	DBpedia	Effectiveness

[a] `http://dblp.uni-trier.de`

[b] `https://developers.google.com/freebase`

[c] `http://dbpedia.org`

[d] `http://data.semanticweb.org/`

[e] `http://www.yovisto.com`

[f] Across 100+ repositories, which provided around 3GB of metadata.

[g] `http://DBTune.org`

[h] Contains books, movies, articles. At `http://lod.sztaki.hu`

[i] `http://yago-knowledge.org`

The related work shows that there are many different approaches to look at exploratory search with semantic data and the table specifically indicates that each of these related works focuses on a very specific aspect about exploring (linked) data. Furthermore, it shows the various aspects and possible datasets/domains of application. This work is an alternative way explore data, with its own focus on the visualization of relationships between resources, tested with DBLP, DBpedia, and data from social media. There is a strong emphasis on efficiency and effectiveness in the evaluation: both from a user perspective and from an information retrieval perspective. Rather than focusing on top-k results or faceting, the presented experiments look into the trade-offs in terms of serendipity while exploring data on the web.

Semantic Search

Many different concepts and definitions for semantic search exist [7, 11, 16, 39]. The understanding of semantic search in the scope of information retrieval (IR) [6] differs in many aspects from the one in the Semantic Web community [32]. However, common to all semantic search approaches is the use of a semantic model which includes describing resources using controlled vocabularies, a query - and a matching framework.

Hermes [33] translates the keywords into structured queries while our approach tries to satisfy the user needs by expanding the results using the paths within the connected linked data graphs in the context of the user's social profile upon which the user can than expand or refine to context over several iterations. The experimental user interface requires a special domain knowledge to get useful information. Hermes' core consists of query disambiguation and of distributed query processing. An alternative is *Poweraqua* [18, 19], a query answering system which does not assume that the user has any prior information about the underlying semantic structure or resources. Relation similarities are determined and triples are linked by expressing the input query as ontology concepts after identifying and mapping the terminology using a dedicated service.

Instead of working on top of well-structured datasets, web crawling engines for the semantic web followed hyperlinks through documents whose annotations were indexed and delivered classical lists as results, for instance *SWSE* [15] and *Sindice* [26]. Other engines added support for top-k queries and allow matching keywords within attributes and relations in the RDF data to improve scoring functions based on textual relevancy and relationship popularity [17]. Well-defined SPARQL fragments were identified that can be naturally captured using faceted search as a query paradigm [1] for the development of *SemFacet*, a faceted search interface for exploratory search. In a typical exploratory search session users combine keyword-based search with visual feedback allowing to extend the search iteratively based on different aspects, which they can recognize as facets. These facets are similar through what would be exposed in a faceted search. The idea behind presenting the results with different facets is to offer always and at each step an explanation to foster understanding why certain results are showed, or as Waitelonis described it: "Exploratory semantic search is based on generic facets, enabling the user to better refine and broaden search queries and to provide content-based recommendations" [36].

Interactive Search

The *mSpace* framework and architecture is a platform to deploy lightweight Semantic Web applications where foreground associative interaction is one of first such interfaces [30] and linked data is not presented as a list or a graph but in parallel tabs. Other related work emphasized more the aspect of the relationship exploration. As noted by Heim et al. interactive exploration is only possible with a human involved, since only a user can judge whether a found relationship is relevant in a certain situation or not. In their work they presented an approach for the interactive discovery of relationships between selected elements via the Semantic Web [14] and implemented the *RelFinder* [13] as a proof-of-concept. Another related graph exploration tool that maximally exploits the linkedness of linked data is *LODMilla* [24].

The resources users discover along the paths among resources encountered during search is becoming 'a destination' on its own. *Waitelonis et al.*, who investigated the analysis and cleansing of linked data resources on structural, semantical level because they found publicly available linked datasets often did not meet quality requirements, concluded [36]: "by harnessing the meaning of content associative, faceted, and exploratory search interfaces can be developed providing high quality search results (by means of recall and precision) ... shifting to an exploratory approach, web search is becoming a quest for knowledge, guiding the user along new pathways to serendipitous findings".

The *Discovery Hub*, an exploratory search system supporting faceted browsing of search results, enables the exploratory search tasks by drawing attention to resources and associations that convey a lot of knowledge regarding the users' initial interest and leverages linked data richness to explore topics of interest over DBpedia through several perspectives [22, 23].

A study on an exploratory semantic browser applied to the musical domain, *Pinta*, indicated that semantic facets support exploratory search and facilitate serendipitous learning and confirmed that the overview of a knowledge structure presented with classification level tags is beneficial for the success of analytical tasks [10]. *Aemoo* was implemented leveraging ontology patterns for data exploration and integrated data coming from heterogeneous data sources [25].

2.2 Principles

Exploratory search covers a broader class of tasks than typical information retrieval where new information is sought in a bounded conceptual area rather than having a specific goal in mind. The users' demand to discover data across a variety of sources at once, requires searching facilities adaptive to their adjustments while they discover the data that were just put at their disposal.

In general exploratory search describes either the problem context that motivates the search or the process by which the search is conducted [20]. This means that the users start from a vague but still goal-oriented defined information need and are able to refine their need upon the availability of new information to address it, with a mix of keyword look-up, expanding or rearranging the search context, filtering and analysis. Such queries will start simple but become more complicated as users get more and more familiar with the data after a while. The resolution of vague or complex information problems requires exploratory behaviors, for instance: multiple publishers providing resources. During exploratory search and analysis, it is likely that the problem context becomes better understood, allowing the searchers to make more informed decisions about interaction or information use [38].

Rather than attempting a direct search and then immediately jumping to the (final) result, the observed advantages of searching by taking small steps include that it allowed users to specify less of their information need and provided a context in which to understand their results [31]. During exploratory searches, it is likely that the problem context becomes better understood, allowing users to make more informed decisions about interaction or information use [38].

Aula and Russel made a distinction between complex and exploratory search tasks [2]: "... exploratory search may sometimes be complex, but is not necessarily so, and is characterized more accurately by the degree of clarity the searcher has about the goal. Complex search tasks often include exploring the topic, but do not necessarily require exploration or may require exploration only in certain phases of the search process." They suggested that complex search tasks with an unclear initial goal are the ones where current search tools do not offer sufficient support. From a system survey on linked data exploration systems [21], it was observed that massive use of linked data based exploratory search functionalities and systems constitutes an improvement for the evolving web search experience and this tendency is enhanced by the observation that users are getting more and more familiar with structured data in search through the major search engines.

2.3 Research Questions and Hypotheses

We investigate how users explore information and gain insights through applications that enable them to interact with distributed heterogeneous data sources. The following questions are addressed for attaining a set of techniques for exploratory search:

RQ1 *Can exploratory search efficiently and adequately address the user's intent when revealing relationships between resources?*

RQ2 *To what degree do users' search actions influence the relevance and precision of search results?*

RQ3 *How does a justification of the presented results influence the user's certainty in getting closer to achieving the search goal?*

RQ4 *How do users gradually refine a search query by interacting with its search results?*

It is relevant to measure if and how well agreeing on semantics proves to be useful in tackling these issues. Our approach and evaluation illustrates how to apply semantic paradigms for search, exploration and querying.

The research questions induce the following hypotheses:

HYP1 Interaction with the result set makes the information contained in the initial search query more specific, leading to more and more specific queries, targeted towards the search goal.

HYP2 When exploring the data, indications such as facets, visualizations (charts, graphs etc.) reduce the number of steps to achieve a search goal.

HYP3 The way search results are ordered affects the precision but does not affect the search process, for example in terms of the number of steps needed to reach the search goal.

The research questions and hypotheses posed here are addressed in the following chapters.

- Chapter 4: **RQ2**, **RQ4** and HYP2.
- Chapter 5: **RQ1**, **RQ4**, HYP1 and HYP3.
- Chapter 6: **RQ1** and **RQ3**.
- Chapter 8: all research questions and hypotheses, applied to the use case on research exploration explained in Chapter 7.

References

[1] M. Arenas, B. C. Grau, E. Kharlamov, Š. Marciuška, and D. Zheleznyakov. Faceted search over rdf-based knowledge graphs. *Web Semantics: Science, Services and Agents on the World Wide Web*, 37:55–74, 2016.

[2] A. Aula and D. M. Russell. Complex and exploratory web search. *Information Seeking Support Systems Workshop (ISSS 2008), Chapel Hill, NC, USA*, 2008.

[3] M. J. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online review*, 13(5):407–424, 1989.

[4] M. J. Bates. Where should the person stop and the information search interface start? *Information Processing & Management*, 26(5):575–591, 1990.

[5] A. van den Bosch, T. Bogers, and M. de Kunder. Estimating search engine index size variability: a 9-year longitudinal study. *Scientometrics*, 107(2):839–856, 2016.

[6] P. Castells, M. Fernandez, and D. Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):261–272, 2007.

[7] J. Chu-Carroll, J. M. Prager, K. Czuba, D. A. Ferrucci, and P. A. Duboue. Semantic search via XML fragments: a high-precision approach to IR. E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Jorvelin, editors, *SIGIR*, pages 445–452, ACM, 2006.

[8] S. J. Clarke and P. Willett. Estimating the recall performance of web search engines. *Aslib Proceedings*. Volume 49 of 7. MCB UP Ltd, pages 184–189, 1997.

[9] L. De Vocht, S. Softic, E. Mannens, M. Ebner, and R. Van de Walle. Aligning web collaboration tools with research data for scholars. *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*. WWW Companion 2014, pages 1203–1208, International World Wide Web Conferences Steering Committee, Seoul, Korea, 2014.

[10] V. Dimitrova, L. Lau, D. Thakker, F. Yang-Turner, and D. Despotakis. Exploring exploratory search: a user study with linked semantic data. *Proceedings of the 2nd International Workshop on Intelligent Exploration of Semantic Data*. ACM, pages 2, 2013.

[11] R. V. Guha, R. McCool, and E. Miller. Semantic search. *WWW*, pages 700–709, 2003.

[12] A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. *Special interest tracks and posters of the 14th international conference on World Wide Web*. ACM, pages 902–903, 2005.

[13] P. Heim, S. Hellmann, J. Lehmann, S. Lohmann, and T. Stegemann. Relfinder: revealing relationships in rdf knowledge bases. *Proceedings of the 4th International Conference on Semantic and Digital Media Technologies (SAMT 2009)*. SAMT '09, pages 182–187, Springer, Berlin, Heidelberg, 2009.

[14] P. Heim, S. Lohmann, and T. Stegemann. Interactive relationship discovery via the semantic web. In: *The Semantic Web: Research and Applications*, pages 303–317. Springer, 2010.

[15] A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, and S. Decker. Searching and browsing linked data with SWse: the semantic web search engine. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):365–401, 2011.

[16] A. Kiryakov, B. Popov, D. Ognyanoff, D. Manov, A. Kirilov, and M. Goranov. Semantic annotation, indexing, and retrieval. D. Fensel, K. P. Sycara, and J. Mylopoulos, editors, *International Semantic Web Conference*. Volume 2870 of Lecture Notes in Computer Science, pages 484–499, Springer, 2003.

[17] H. Li. An approach to semantic information retrieval. *Cloud and Service Computing (CSC), 2012 International Conference on*. IEEE, pages 161–167, 2012.

[18] V. Lopez, M. Fernández, E. Motta, and N. Stieler. Poweraqua: supporting users in querying and exploring the semantic web. *Semantic Web*, 3(3):249–265, 2012.

[19] V. Lopez, E. Motta, and V. Uren. Poweraqua: Fishing the semantic web. Springer, 2006.

[20] G. Marchionini. Exploratory search: from finding to understanding. *Commun. ACM*, 49(4):41–46, April 2006.

[21] N. Marie and F. L. Gandon. Survey of linked data based exploration systems. *Proceedings of the 3rd International Workshop on Intelligent Exploration of Semantic Data (IESD 2014) co-located with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Italy, October 20, 2014.* 2014.

[22] N. Marie, F. L. Gandon, A. Giboin, and É. Palagi. Exploratory search on topics through different perspectives with dbpedia. *Proceedings of the 10th International Conference on Semantic Systems, SEMANTICS 2014, Leipzig, Germany, September 4-5, 2014*, pages 45–52, 2014.

[23] N. Marie, F. Gandon, M. Ribière, and F. Rodio. Discovery hub: on-the-fly linked data exploratory search. *Proceedings of the 9th International Conference on Semantic Systems*. I-SEMANTICS '13, pages 17–24, ACM, Graz, Austria, 2013.

[24] A. Micsik, Z. Tóth, and S. Turbucz. Lodmilla: shared visualization of linked open data. *International Conference on Theory and Practice of Digital Libraries*. Springer, pages 89–100, 2013.

[25] A. G. Nuzzolese, V. Presutti, A. Gangemi, S. Peroni, and P. Ciancarini. Aemoo: linked data exploration based on knowledge patterns. *Semantic Web*, 2017.

[26] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice.com: a document-oriented lookup index for open linked data. *Int. J. of Metadata and Semantics and Ontologies*, 31:37–52, 2008.

[27] S. Petridou, G. Pallis, A. Vakali, and G. P. A. Pomportsis. Web data accessing and the web searching process. *ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2003), Tunis, Tunisia*, 2003.

[28] D. M. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[29] C. J. V. Rijsbergen. Information Retrieval. 2nd. Butterworth-Heinemann, Newton, MA, USA, 1979.

[30] m. c. Schraefel, D. A. Smith, A. Owens, A. Russell, C. Harris, and M. Wilson. The evolving mspace platform: leveraging the semantic web on the trail of the memex. *Proceedings of the Sixteenth ACM Conference on Hypertext and Hypermedia*. HYPERTEXT '05, pages 174–183, ACM, Salzburg, Austria, 2005.

[31] J. Teevan, C. Alvarado, M. S. Ackerman, and D. R. Karger. The perfect search engine is not enough: a study of orienteering behavior in directed search. *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, pages 415–422, 2004.

[32] T. Tran, D. M. Herzig, and G. Ladwig. Semsearchpro - using semantics throughout the search process. *Web Semantics: Science, Services and Agents on the World Wide Web.* 9(4):349–364, 2011.

[33] T. Tran, H. Wang, and P. Haase. Hermes: dataweb search on a pay-as-you-go integration infrastructure. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3), 2009.

[34] P. Vakkari. A theory of the task-based information retrieval process: a summary and generalisation of a longitudinal study. *Journal of documentation*, 57(1):44–60, 2001.

[35] P. Vakkari and E. Sormunen. The influence of relevance levels on the effectiveness of interactive information retrieval. *Journal of the Association for Information Science and Technology*, 55(11): 963–969, 2004.

[36] J. Waitelonis, M. Knuth, L. Wolf, J. Hercher, and H. Sack. The path is the destination–enabling a new search paradigm with linked data. *Proceedings of the Workshop on Linked Data in the Future Internet at the Future Internet Assembly*, 2010.

[37] R. W. White, G. Muresan, and G. Marchionini. Evaluating exploratory search systems. *Proceedings of the ACM SIGIR 2006 Workshop on Evaluating Exploratory Search Systems*, pages 1–2, 2006.

[38] R. W. White and R. A. Roth. Exploratory search: beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1):1–98, 2009.

[39] L. Zhang, Y. Yu, J. Zhou, C. Lin, and Y. Yang. An enhanced model for searching in semantic portals. A. Ellis and T. Hagino, editors, *WWW*, pages 453–462, ACM, 2005.

Chapter 3

Linked Data on the Web

> *We're entering a new world in which data may be more important than software.*
>
> —Tim O'Reilly.

This chapter summarizes the key concepts behind linked data and in particular querying linked data. We sketch briefly the origin and give a clear definition of each concept.

3.1 The Semantic Web

One of the main motivations behind the Semantic Web, a vision of Tim Berners-Lee et al. [3, 4] was bootstrapping a web of intelligent machines. The machines are software agents which carry out sophisticated tasks such as intelligent search. Several definitions were examined to capture the essence of 'agent' in a formal definition to allow a clear distinction between a software agent and an arbitrary program [15]. Note though that the 'intelligent agents' intended in this context for example do not try to do everything *for* a user or as Hendler posed it with a travel agent analogy: "... the agents would find possible ways to meet user needs and offer the user choices for their achievement; much as a travel agent might give you a list of several flights to take, or a choice of flying as opposed to taking a train, a Web agent could offer several possible ways to get you what you need on the Web" [19]. He claimed that this vision on intelligent agents "is quite compelling and many people now *(in 2001, author's note)* believe they *(intelligent agents, author's note)* will be necessary if we are ever to tame the increasing complexities caused by the accelerating and virtually uncontrolled growth of the World Wide Web" [20]. The semantic web and its data described following semantic models provides a huge "global database" for knowledge based applications. It attempts to adapt information access by addressing both users and machines.

Open World versus Closed World Assumption

There are two ways to treat databases: following an 'open world' or a 'closed world' assumption. This distinction was introduced by Reiter [23]: "... the open world assumption, assumes only the information given in the database and hence requires all facts, both positive and negative, to be explicitly represented; under the open world assumption, –gaps– in one's knowledge about the domain are permitted". To derive a negative fact from a database under the closed world assumption, one attempts to prove the positive fact true; if one fails to prove the positive fact, then the negative data is assumed to be true [22].

Plurality and contradictions cannot be excluded on the semantic web. A database traditionally presents one view on the 'truth', while the semantic web may have multiple possible worlds, where each world represent a view in which one truth is represented. This is because the semantic web, as a global database, has an open world assumption. Two of the most essential semantic web building blocks explicitly declared this [9]:

(i) the Resource Description Framework (RDF): "RDF is an open-world framework that allows anyone to make statements about any resource; in general, it is not assumed that complete information about any resource is available"[1].

(ii) the Web Ontology Language (OWL): "OWL makes an open world assumption; that is, descriptions of resources are not confined to a single file or scope; while class C1 may be defined originally in ontology O1, it can be extended in other ontologies ... new information cannot retract previous information; new information can be contradictory, but facts and entailments can only be added, never deleted."[2]

[1] https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

[2] https://www.w3.org/TR/owl-guide/

Resources

The term 'resource' has been mentioned in the previous chapters. The term was first introduced to refer to special pages and content within a webpage[3], more specifically the destination of a Uniform Resource Locator (URL), for example: someone's contact information page. Later, the definition was extended to any piece of information that can be pointed at[4], not only web pages, such as the geo-location of an address. The W3C Web Architecture[5] distinguishes between two types of resources:

(i) **Information resources**[6]: things that may have a digital representation (on the web) such as data, web-services, ontologies, and documents.

(ii) **Non-information resources**[7]: things, concepts and events that do not have a representation on a machine but where their description might have (e.g. in a document). For example: a meeting report is a document (information resource) but the meeting itself is a non-information resource.

Unique Identification of Resources on the Web

Data on the Web represented as linked data are uniquely identified by a string of characters, a Universal Resource indicator (URI). A URI is a formal way to refer to a resource. The most well-known form of a URI is a URL, which can be seen as equivalent to an address for a webpage. The World Wide Web Consortium (W3C) best practices for linked data[8] expect that URIs follow the hypertext transfer protocol (HTTP) protocol[9]. URIs should be resolvable, to be able to retrieve the content they refer to. This happens either by answering to the URI directly or by following redirects. Unique identification of resources enables interaction with their representations on the Web.

According to W3C guidelines for so-called "cool URIs for the Semantic Web" it is important that URIs make a distinction between "a thing (which may exist outside

[3] RFC3986. https://tools.ietf.org/html/rfc3986

[4] RFC3987. https://tools.ietf.org/html/rfc3987

[5] https://www.w3.org/TR/webarch/

[6] https://www.w3.org/TR/webarch/#id-resources

[7] https://www.w3.org/2001/tag/doc/httpRange-14/2007-08-31/HttpRange-14#iddiv193805720

[8] https://www.w3.org/TR/ld-bp/#HTTP-URIS

[9] RFC2616, https://www.ietf.org/rfc/rfc2616

the web) and a web document describing the thing”[10]. This corresponds to the two types of resources: non-information and information resources.

Resource Description Framework

The Resource Description Framework (RDF) [21] is a graph based representation of linked data instances. It is a method for conceptual description or modeling of information on the Web[11]. Around 2000, when RDF was slowly gaining popularity, it had to compete with XML as technology for interoperability. The main motivation behind RDF was the notion that “XML and RDF are the current standards for establishing semantic interoperability on the Web, but XML addresses only document structure; RDF better facilitates interoperation because it provides a data model that can be extended to address sophisticated ontology representation techniques” [10].

The base unit of expression in RDF is a ‘triple’: a statement containing *subject*, *predicate* and *object*. For example: *Tim Berners-Lee*, *birthPlace*, *London*; meaning literally as stated, Tim Berners-Lee’s birthplace (is) London. Essentially, every component that is a subject or a predicate is a URI, such as *dbo:birthPlace*. The advantage is that every subject and predicate can be uniquely identified on the Web. Objects can be a URI but also literally represented. Furthermore every request to a URI may result in more triples, thus more RDF data in one of its representations. Triples may be extended with a name (also a URI) of the graph they belong to, each statement then consists of four elements: *subject*, *predicate*, *object* and *graph*; this is called a ‘quad’.

RDF has multiple representations recommended by the W3C. The representations either support RDF datasets with a single unnamed graph (triples) or multiple named graphs (quads).

The following representations focus on triples:

- **RDF/XML**: an XML syntax for RDF;
- **N-Triples**: N-Triples is a line-based, plain text format for encoding an (single) RDF graph[12];

[10] https://www.w3.org/TR/cooluris/#distinguishing
[11] https://www.w3.org/TR/rdf-syntax-grammar/
[12] https://www.w3.org/TR/n-triples

- **Turtle**: a superset of N-Triples and a common syntax allowing RDF graphs to be written in a compact and natural text form. Even though RDF is mainly intended for machine interoperability, the popularity of Turtle is mainly due to the human-friendly and easy-to-read syntax[13].

Some representations support ‘named graphs’, sets of RDF triples grouped in one or more graphs identified by a URI, rather than a default graph without a name:

- **JSON-LD**: a JSON-based serialization[14];
- **N-Quads**: as a superset of N-Triples, N-Quads is also a line-based representation format but it adds support for datasets consisting of multiple graphs[15];
- **TriG**: a textual syntax for RDF allowing an RDF dataset to be completely written in a compact and natural text form, with abbreviations for common usage patterns and datatypes[16]. TriG is an extension of Turtle.

RDF can be embedded in HTML:

- **Interleaved**: RDFa is an addition to HTML to support the enrichment of documents with RDF triples[17];
- **Appended**: Inside an HTML *script*-tag RDF content may be included. For example, JSON-LD by setting the script *type* attribute to *application/ld+json*[18].

Vocabularies, Ontologies

The strict definition of a vocabulary is a ‘set or list of words’. The term vocabulary is often used in exchange with the term ontology. The distinction is subtle:

- **Vocabulary**: Building blocks to model data, reusable concepts and properties.
- **Ontology**: A set of concepts *and* their relationships.

[13] `https://www.w3.org/TR/turtle`
[14] `https://www.w3.org/TR/json-ld`
[15] `https://www.w3.org/TR/n-quads/`
[16] `https://www.w3.org/TR/trig/`
[17] `https://www.w3.org/TR/html-rdfa/`
[18] `https://www.w3.org/TR/json-ld/#embedding-json-ld-in-html-documents`

The term vocabulary occurs mostly in less formal contexts while the term ontology occurs mostly in complex and formal context. However, according to W3C there is “no clear division”[19] between vocabulary and ontology.

Certain ontologies are well-known and often reused:

- **RDFS**: RDF Schema[20], base vocabulary to describe other vocabularies;
- **OWL**: The Web Ontology Language[21]. A family of knowledge representation languages. Concepts for detailed vocabularies with strict constraints;
- **SKOS**: Simple Knowledge Organization System[22] Organization of concepts and hierarchies, taxonomies;
- **Dublin Core**: Common meta-data terms.
- **Schema.org**: A common set of schemas for structured data markup on a web page[23]. In practice the schemas are being used for a wider variety of purposes.

Linked Data

Linked Data is a method of publishing structured data so that it can be interlinked and become more meaningful. It builds on standard Web technologies like HTTP and RDF. It does not primarily serve documents and pages for human readers, it shares machine-readable pieces of information. This enables data from different sources to be connected and queried. The developments around linked data lead to the exposure of large amounts of data on the Web eligible for automated processing in software agents [5].

Linked Data follows the principles, as issued by a note of Tim Berners-Lee[24]:

- Use URIs as names for things
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)

[19] https://www.w3.org/standards/semanticweb/ontology
[20] https://www.w3.org/TR/rdf-schema/
[21] https://www.w3.org/TR/owl-features/
[22] https://www.w3.org/TR/skos-reference/
[23] http://schema.org
[24] https://www.w3.org/DesignIssues/LinkedData.html

- Include links to other URIs. so that they can discover more things.

In the same note he also stated that "the Semantic Web is not just about putting data on the web; it is about making links, so that a person or machine can explore the web of data. With linked data, when you have some of it, you can find other, related, data." Linked data can be seen as many RDF statements, a combination of triples, forming a graph on web-scale. The smallest unit there being the URI, uniquely identifying nodes and edges.

From Structured and Unstructured Data to Linked Data

Most data is not available as linked data by default. Even though tools exist to integrate data from distributed heterogeneous sources and convert them to Linked Data, the process as a whole remains complicated [11].

Structured data is typically available in relational databases as tables or spreadsheets. To make this data available in RDF we use two types of processes, Predefined (static) annotations using the API of the resource provider to load the information from the data repository and dynamic mapping between the ontology and the data repository, such as with a tool like D2RQ [6] or Ontop [8] in case of relational databases, or the RDF Mapping Language (RML) [11] for heterogeneous mappings.

Unstructured data can be converted into structured data using natural language processing techniques, mainly named entity recognition. The quality of the entity recognition is influenced by the richness of the ontology [12]. Some important Named Entity Recognition/Linked Data systems are: GATE[25], DBpedia Spotlight[26], Alchemy[27], or Apache Stanbol[28]. They are increasingly applied to automatically generate structured data (entities) from unstructured resources such as Web sites, documents or social media. Each of the tools has their own strengths and weaknesses in regard to the type of extracted named entities and in terms of provided specific (and precise) results given a ground truth "golden standard" established by a test panel [24].

[25] http://gate.ac.uk
[26] http://spotlight.dbpedia.org
[27] http://www.alchemyapi.com
[28] http://stanbol.apache.org

3.2 Query Execution

Selecting specific data from an RDF dataset is done via the SPARQL protocol and query language, a W3C recommendation [18]. SPARQL as a *query language* defines fixed keywords, like other query languages, to select, insert, update or delete data. SPARQL as a *protocol* defines an API for querying RDF datasets over HTTP. An RDF dataset may be exposed on the Web via a SPARQL endpoint. A SPARQL endpoint is a web service allowing the execution of queries sent by client applications through HTTP.

Basic Graph Patterns

A SPARQL query's main composition unit is the 'basic graph pattern' (BGP). Each BGP consists of one or more triple *patterns*. Each pattern can have one or more variables in its components (subject, predicate and/or object). It is the task of the query engine to find solutions that bind to the variables occurring in the BGPs. For example the query in Listing 3.1 asks to *select* triples matching the *outcome* of *Napoleon*'s *commanded* wars with the number of *casualies* and results in the response in Listing 3.1.

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>

SELECT ?war ?outcome ?casualties
WHERE {
  ?war dbo:result ?outcome .
  ?war dbo:commander dbr:Napoleon .
  ?war dbp:casualties ?casualties
}
```

Listing 3.1: Example SPARQL query about the outcomes and casualties of battles with Napoleon as commander.

SPARQL Query Processing

To be able to respond to a SPARQL query, the query engine interprets the query using SPARQL algebra[29], similar as to SQL engines translate SQL queries to SQL algebra. Each query is being translated to a tree-structure. A SPARQL engine, like Jena ARQ[30], will iterate over this tree to resolve the query. Each engine will have its

[29] https://www.w3.org/TR/sparql11-query/#sparqlQuery

[30] https://jena.apache.org/documentation/query

Table 3.1: Results matching the SPARQL SELECT query about Napleon's commanded wars, their outcomes and casualties.

war	outcome	casualties
dbr:Battle_of_Waterloo	Decisive Coalition victory	Total: 41,000* 24,000 to 26,000 killed, wounded including 6,000 to 7,000 captured* 15,000 missing
dbr:Battle_of_Craonne	French victory	5400
dbr:Battle_of_Eckmuhl	French victory	12000
dbr:Battle_of_Landshut_(1809)	French victory	9000
dbr:Ulm_Campaign	Decisive French victory	2000
dbr:Battle_of_Borghetto	French victory	500
(... 283 results)		

own strategy to do this. ARQ, for example, visits the nodes one by one, executing any operations if necessary. Before query processing, engines might tweak and optimize the algebraic structure of the tree for better query performance [25].

Basic operations. The tree in Figure 3.1 represents the SPARQL algebra for the example in Listing 3.1. There are five operations in this tree: **triple**, **bgp**, **prefix**, **project** and **base**. There are more operations in SPARQL, we refer the reader therefore to literature. A good starting point is the W3C recommended SPARQL specification[31].

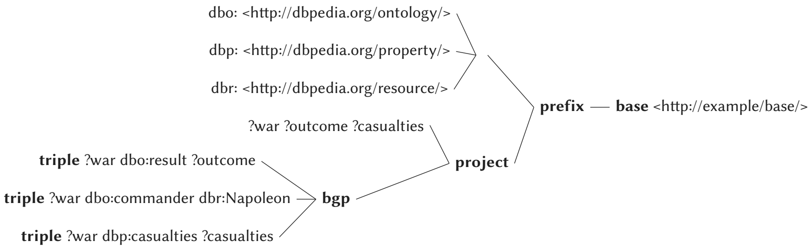

dbo: <http://dbpedia.org/ontology/>
dbp: <http://dbpedia.org/property/>
dbr: <http://dbpedia.org/resource/>
prefix
base <http://example/base/>
?war ?outcome ?casualties
project
triple ?war dbo:result ?outcome
triple ?war dbo:commander dbr:Napoleon
bgp
triple ?war dbp:casualties ?casualties

Figure 3.1: SPARQL Algebra for the example in listing 3.1 with a single BGP consisting of two triple patterns.

The **triple** operation retrieves the matching triples for this node. This operation can only be a leaf, because it can have no children. It takes *subject*, *predicate* and *object* as attributes. The **prefix** operation resolves prefixes within the graph and the **base** operation sets the domain in the response for all results that have relative URIs.

[31] https://www.w3.org/TR/sparql11-query/#sparqlQuery

The **project** operation binds the results to the requested variables, in this case *?war*, *?outcome* and *?casualties*.

An operation that may have one or more children is **bgp**. A *bgp* will join the underlying results of its children. In the example case, it are the triples matching the triple patterns. When unsure whether the military conflict (*?war*) has information about the number of casualties, it might be interesting to include the OPTIONAL keyword, as shown in Listing 3.2. The OPTIONAL keyword indicates that a certain BGP is optional.

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>

SELECT ?war ?outcome ?casualties
WHERE {
  ?war dbo:result ?outcome .
  ?war dbo:commander dbr:Napoleon .
  OPTIONAL { ?war dbp:casualties ?casualties }
}
```

Listing 3.2: Making the casualty count optional.

Leftjoin. Translating the query in Listing 3.2 to SPARQL algebra introduces a **leftjoin** (shown in Figure 3.2) of the two BGPs instead of a single BGP operation.

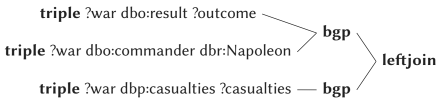

triple ?war dbo:result ?outcome
triple ?war dbo:commander dbr:Napoleon
triple ?war dbp:casualties ?casualties
bgp
bgp
leftjoin

Figure 3.2: SPARQL Algebra for the example in Listing 3.2 with OPTIONAL statement introduces a *leftjoin* of two BGPs. The **base**, **prefix** and **project** operation are not shown here for clarity, but they remain unaffected.

The difference between the **bgp** and the **leftjoin** is that the **bgp** will traverse all underlying triple pattern matches and 'tie' them together based on co-occurring variables. In the example *?war* occurs in both cases. This means that any result after the **bgp** operation will have to occur as subject in the triple patterns of *dbo:result*, *dbo:commander* and as subject of *dbp:casualties* in the third triple pattern. With the **leftjoin** all results from the first, the 'left', child, will be enriched with results from the second, 'right', child. All additional triples where the results for the common variables, in this case *?war*, overlap will be added. The left child corresponds to the **bgp** with two triple patterns: *dbo:result* and *dbo:commander*; and the right child

Table 3.2: Results matching the SPARQL SELECT query about Napleon's commanded wars their outcome with the number of casualties optional.

war	outcome	casualties
dbr:Battle_of_Waterloo	Decisive Coalition victory	Total: 41,000* 24,000 to 26,000 killed, wounded including 6,000 to 7,000 captured* 15,000 missing
dbr:Battle_of_Craonne	French victory	5400
dbr:Battle_of_Eckmühl	French victory	12000
(...)		
dbr:War_of_the_Fourth_Coalition	French victory,Treaties of Tilsit	
dbr:Mediterranean_campaign_of_1798	Allied victory	
dbr:French_campaign_in_Egypt_and_Syria	Ottoman-British victory	
(... 337 results)		

to the **bgp** with a single triple pattern about the *dbp:casualties*. The results will contain the same results as in Table 3.1 and also contain additional results where no information on the casualties is present (empty cells), as shown in Table 3.2.

Property paths. One may be interested in finding out which military conflicts these wars and battles commanded by Napoleon may belong to, according to DBpedia. One way to do this, would be to explicitly ask for the military conflict information using a query, as shown in Listing 3.3. The SELECT query results in Table 3.3. The results indicate that some of the wars and battles belong to a military conflict which in turn belongs to the Napoleonic Wars. This means that some wars and battles are indirectly related.

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?war ?conflict
WHERE {
  ?war dbo:commander dbp:Napoleon .
  ?war dbo:isPartOfMilitaryConflict ?conflict
}
```

Listing 3.3: Listing the wars and battles commanded by Napoleon with military conflicts they belong to.

A way to include the indirect relations, is to use SPARQL property paths[32], for example to be sure to include all wars and battles commanded by Napoleon during the Napoleonic wars. SPARQL property paths allow chaining one or more predicates to retrieve indirectly related nodes bound to these predicates.

[32] https://www.w3.org/TR/sparql11-property-paths

Table 3.3: List of wars and battles commanded by Napoleon. We note that for example the *Battle of Craonne* is part of the *War of the Sixth Coalition* which belongs to the *Napoleonic Wars*. This does not mean that this is the case for all the battles, for example *Quasi-War* belongs to the *French Revolutionary Wars* which took not place during the *Napoleonic Wars* but in the decade before.

war	conflict
dbr:Battle_of_Waterloo	dbr:Waterloo_Campaign
dbr:Battle_of_Craonne	dbr:War_of_the_Sixth_Coalition
dbr:Battle_of_Eckmühl	dbr:War_of_the_Fifth_Coalition
dbr:Battle_of_Ceva	dbr:French_Revolutionary_Wars
dbr:Quasi-War	dbr:French_Revolutionary_Wars
dbr:Waterloo_Campaign	dbr:Hundred_Days
dbr:Hundred_Days	dbr:Napoleonic_Wars
dbr:War_of_the_Fifth_Coalition	dbr:Napoleonic_Wars
dbr:War_of_the_Sixth_Coalition	dbr:Napoleonic_Wars
(...)	

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>

CONSTRUCT { ?war dbp:result ?outcome }
WHERE {
  ?war dbo:result ?outcome .
  ?war dbo:commander dbr:Napoleon .
  ?war dbo:isPartOfMilitaryConflict dbr:Napoleonic_Wars
}
```

Listing 3.4: Example SPARQL query about the result of battles with Napoleon as commander and part of the Napoleonic Wars.

```
@prefix dbo:  <http://dbpedia.org/ontology/> .
@prefix dbr:  <http://dbpedia.org/resource/> .

dbr:War_of_the_Sixth_Coalition  dbo:result  "Coalition victory,Treaty of Fontainebleau,First
      Treaty of Paris" .
dbr:War_of_the_Fifth_Coalition  dbo:result  "French victory,Treaty of Schonbrunn" .
dbr:War_of_the_Fourth_Coalition dbo:result  "French victory,Treaties of Tilsit" .
dbr:War_of_the_Third_Coalition  dbo:result  "French victory,Treaty of Pressburg" .
dbr:French_invasion_of_Russia   dbo:result  "Destruction of French Allied Army" .
dbr:Haitian_Revolution    dbo:result  "Haitian victory" .
dbr:Peninsular_War    dbo:result  "Treaty of Paris" .
dbr:Hundred_Days    dbo:result  "Coalition victory,Second Treaty of Paris" .

# (... 34 statements)
```

Listing 3.5: Triples matching the SPARQL CONSTRUCT query about Napleon's commanded wars outcome.

The example of Listing 3.4 changes to the query in Listing 3.6, adding a + operator to the *dbo:isPartOfMilitaryConflict* property, the + indicates one or more occurrences. A snippet from the results of the query in Listing 3.6 is given in Listing 3.7. We note that there are 109 statements in the results compared to 34 in the results in Listing 3.5 of the query in Listing 3.4 without the property path.

In this case the predicate is *fixed* and not a variable. An arbitrary number of variable predicates could lead to an explosion of possible matching results. Chapter 6 will explain a heuristically optimized solution for this.

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>

CONSTRUCT { ?war dbp:result ?outcome }
WHERE {
  ?war dbo:result ?outcome .
  ?war dbo:commander dbr:Napoleon .
  ?war dbo:isPartOfMilitaryConflict+ dbr:Napoleonic_Wars
}
```

Listing 3.6: Example SPARQL query about the result of battles with Napoleon as commander indirectly part of the Napoleonic Wars.

```
@prefix dbo:  <http://dbpedia.org/ontology/> .
@prefix dbr:  <http://dbpedia.org/resource/> .

dbr:Battle_of_Waterloo  dbr:result  "Decisive Coalition victory" .
dbr:Battle_of_Craonne dbr:result  "French victory" .
dbr:Battle_of_Eckmuhl dbr:result  "French victory" .
dbr:War_of_the_Sixth_Coalition  dbr:result  "Coalition victory,Treaty of Fontainebleau,First
      Treaty of Paris" .
dbr:Battle_of_Landshut_(1809) dbr:result  "French victory" .
dbr:Ulm_Campaign  dbr:result  "Decisive French victory" .
dbr:War_of_the_Fifth_Coalition  dbr:result  "French victory,Treaty of Schonbrunn" .
dbr:Waterloo_Campaign dbr:result  "Coalition victory,Second Treaty of Paris" .

# (... 109 statements)
```

Listing 3.7: Triples matching the SPARQL CONSTRUCT query about Napleon's commanded wars indirectly part of the Napoleonic Wars and their result.

Resolving triples. To resolve a **triple** operation, the query engine depends on the underlying index structure of the triples. One way to store the triples is using a 'triple store' (or RDF store) like Virtuoso[33] or Blazegraph[34]. The SPARQL query engines of these stores have optimized their underlying data structures and indexes. Other engines like RDF for Java (RDF4J)[35] and Jena[36] are more generic and may work on different data structures using an Open Database Connectivity (ODBC) coupling or a custom API. Another way is to use compression such as for example

[33] https://virtuoso.com/
[34] https://blazegraph.org/
[35] http://rdf4j.org/
[36] https://jena.apache.org/

Header Dictionary Triples (HDT) [14]. HDT is a compressed binary file-format which supports only triple patterns, but enriched with a SPARQL query engine full read-only SPARQL support is available[37].

Figure 3.3 shows the typical architecture of a SPARQL query infrastructure. SPARQL queries are transformed to a SPARQL algebra representation which may be optimized. Following the optional optimization, a query plan will be generated that resolves the SPARQL query (via its algebraic tree representation). A query execution plan can be straightforward to complex, the order of which branches in the tree are resolved has an impact on the execution time.

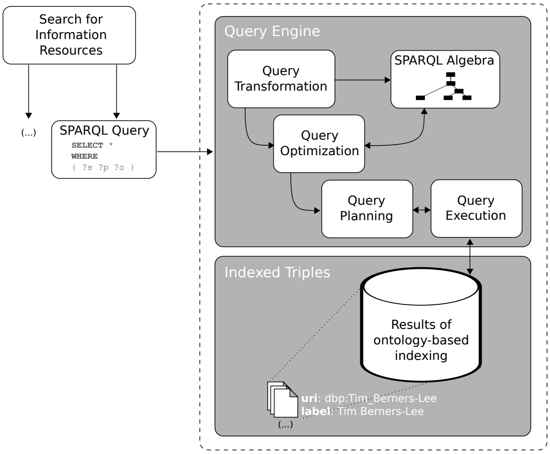


Figure 3.3: A typical SPARQL query infrastructure.

Client vs. Server Trade-offs

SPARQL queries can have varying structures of any complexity depending on the user or application requirements. This large degree of freedom for querying knowledge graphs on the Web can be made possible with:

- **SPARQL endpoints**: the processing load is entirely on the server;
- **Data dumps**: the client processes the data as desired without loading the server, except while downloading the dumps.

The above two methods are extremes when it comes to making data available, regardless of the data's purpose.

[37] http://www.rdfhdt.org/manual-of-hdt-integration-with-jena/

It was found that the majority of published knowledge graphs are not easy to query [13]. On top of that, the mileage may vary when it comes to knowledge graphs that are published queryable as a public SPARQL endpoint: LODStats[38] gives good insight in the availability status of different public SPARQL endpoints [1]. This varying availability is an issue to build any kind of web application, including search applications. In Figure 3.3, there is a distinction between the query engine and the triple index. How (tightly) the two are coupled greatly affects performance, in particular how the communication is established between both, the bandwidth consumption and the CPU and memory use during query processing. In a SPARQL endpoint both the query engine and the triple index are on the same machine, often within the same triple store. When a query engine is used to query multiple remote, distributed SPARQL endpoints, this is called 'federated (SPARQL) querying' [26]. Other interfaces than SPARQL endpoints are also possible, for example Linked Data interfaces based on hypermedia links and controls [27].

Most use cases need more flexible trade-off options, the two extremes do not suffice: downloading data dumps means a huge data overhead – often requiring the set-up of local SPARQL endpoints anyway – and it is too unreliable to do remote querying on SPARQL endpoints via the Web because of the uncertainty about their availability [7]. 'Triple Pattern Fragments' (TPF) is an architectural solution to this by providing self-descriptive hypermedia and being straightforward to maintain [28]. TPFs allow clients to query for triple patterns at the server [29]. TPFs are a type of 'Linked Data Fragments' (LDF). LDF is a way to balance the load between client and server by working with several gradations of pre-defined supported 'fragments' besides supporting the all or nothing granularity (SPARQL vs. data-dump). Strictly speaking, a data dump and SPARQL-endpoint can be seen as a linked data fragment as well, respectively the largest (all data) versus the most flexible (any kind of query). Dereferencing URIs can be seen as another type of fragment: a document with triple statements about a particular *subject*. When putting these fragments on a horizontal axis expressing the workload more towards client and server we get Figure 3.4. The axis goes from generic (left) to very specific queries (right).

In theory, both client side and server side processing of SPARQL queries allow the same applications. Client side processing is particularly useful for federated querying, as the client has to do remote retrieval of results in any case. It has the advantage that multiple resources can be queried at once (using SPARQL). One of the disadvan-

[38] http://stats.lod2.eu/rdfdocs

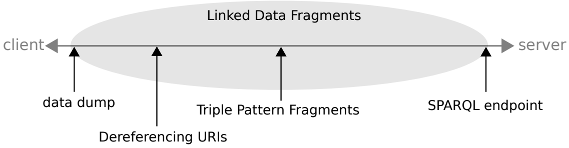


Figure 3.4: Linked Data Fragments organized on an axis expressing the workload trade-off more towards server or client [28].

tages of client side querying processing is the lack of implementations at the moment for analytical queries (counts, aggregates, ...). For these kind of scenarios server side processing is better suited as all data is gathered centrally which allows more specific optimizations, in particular for analytical queries. But the disadvantage of a server side approach is its limited scalability besides increasing the amount of resources. A server could be well functioning for a certain application, but for example when opening up the data for external use, there is less control about the nature or the amount of incoming queries. In those scenarios a client side approach would be advised.

3.3 Link with Exploratory Search

The ideas of exploratory search in combination with the principles of linked data querying align with each other when it comes to knowledge representation and indexing data. To expand and refine search results and to enable revealing paths between results, both queries and results should be aligned to semantic entities that are interlinked by content based relationships. This facilitates extending "the search scope by the option to investigate the semantic context, different time references, or geographical references that are related to the search query or to the original search results." [30]. The data and structure of an indexing language or a knowledge representation should not only be the basis for indexing and searching, but also support navigational purposes and thematic exploration [17]. To facilitate exploratory web search in dealing with a large variety in types of resources, the RDF representations and common vocabularies such as OWL and SKOS come into play. This results in the model contributed by Gödert, outlined in figure 3.5, and is referred to as *ontology-based indexing and retrieval* [16].

Central in Gödert's model is a symbolization of the search index (middle cylinder) of information resources (of which the indexing process itself is represented in the

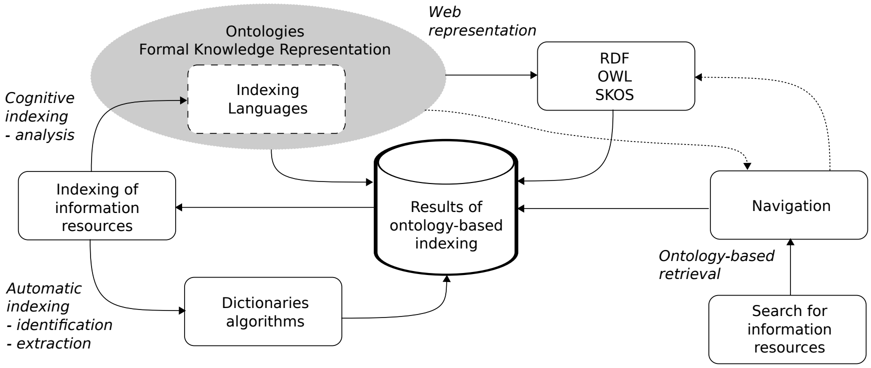


Figure 3.5: Gödert's ontology-based model for indexing and retrieval [16].

block on the left), backs navigation, supports algorithms to operate on it, and aligns indexing languages with a formal knowledge representation with a web counterpart in RDF (top left and top right blocks respectively). There are many ways to develop indexing languages and their relationship types, the internationally standardized way is to follow ISO 25964[39], the standard for thesauri and interoperability with other vocabularies. Indexing languages comprise different types of taxonomies, classification schemes and each type has their own kind of elements. In the model, formal knowledge representation corresponds to the concept of ontology as we defined it. It is important to note that there is a boundary between the cognitive interpretation of concepts and the way it is formalized in an ontology. Dictionaries and algorithms may benefit from automated indexing (identification and extraction of resources), shown in the bottom-left. However, the formal representation itself is the result of a cognitive analysis process based on given information resources (shown in the top-left).

Baeza-Yates et al. stated that "search engines are hindered by their limited understanding of user queries and the content of the Web, and therefore limited in their ways of matching the two" [2]. A formal knowledge representation, an ontology, allows semantic modeling of the human cognition of real-world entities (non-information resources), and representing the descriptive content in information resources on the Web, which should lead to better matching a user need and Web contents. The formal representation is necessary to allow machine processing of information resources. This includes linking data and querying the data using SPARQL.

[39] ISO 25964, `http://www.niso.org/schemas/iso25964`

Concluding Remarks

In this chapter, we explained exploratory search and the Web from a semantic perspective, starting with the definition of the Semantic Web and linked data. Through the Semantic Web's architectural building blocks: resources, URIs, RDF, vocabularies and ontologies; we identified the basics for executing linked data queries with SPARQL. Furthermore, the link between exploratory search and querying linked data is deeply rooted as they are both relying on the result of an ontology-based indexing process, which is a representation of triples optimized for a certain search and query purpose. This optimization can be very generic or application specific and there is a trade-off to be made where the processing workload is placed: more towards clients or more towards servers. The semantic building blocks and link to the index structure is fundamental for the techniques explained and applied in the remainder of this PhD.

References

[1] S. Auer, J. Demter, M. Martin, and J. Lehmann. Lodstats–an extensible framework for high-performance dataset analytics. *International Conference on Knowledge Engineering and Knowledge Management*. Springer, pages 353–362, 2012.

[2] R. Baeza-Yates, M. Ciaramita, P. Mika, and H. Zaragoza. Towards semantic search. *International Conference on Application of Natural Language to Information Systems*. Springer, pages 4–11, 2008.

[3] T. Berners-Lee. Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web. Paperback. HarperCollins, San Francisco, CA, 2000.

[4] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[5] C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee. Linked data on the web (ldow2008). *Proceedings of the 17th international conference on World Wide Web*. WWW '08, pages 1265–1266, ACM, Beijing, China, 2008.

[6] C. Bizer and A. Seaborne. D2rq-treating non-rdf databases as virtual rdf graphs. *Proceedings of the 3rd international semantic web conference (ISWC2004)*. Volume 2004 of Citeseer Hiroshima, 2004.

[7] C. Buil-Aranda, A. Hogan, J. Umbrich, and P.-Y. Vandenbussche. Sparql web-querying infrastructure: ready for action? *International Semantic Web Conference*. Springer, pages 277–293, 2013.

[8] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, and G. Xiao. Ontop: answering SPARQL queries over relational databases. *Semantic Web*, 8(3):471–487, 2017.

[9] J. J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named graphs, provenance and trust. *Proceedings of the 14th international conference on World Wide Web*. ACM, pages 613–622, 2005.

[10] S. Decker, S. Melnik, F. Van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks. The semantic web: the roles of xml and rdf. *IEEE Internet computing*, 4(5):63–73, 2000.

[11] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. Van de Walle. RML: a generic language for integrated rdf mappings of heterogeneous data. *Proceedings of the 7th Workshop on Linked Data on the Web (LDOW2014), Seoul, Korea*, 2014.

[12] D. W. Embley, D. M. Campbell, R. D. Smith, and S. W. Liddle. Ontology-based extraction and structuring of information from data-rich unstructured documents. *Proceedings of the Seventh International Conference on Information and Knowledge Management*. CIKM '98, pages 52–59, ACM, Bethesda, Maryland, USA, 1998.

[13] I. Ermilov, M. Martin, J. Lehmann, and S. Auer. Linked open data statistics: collection and exploitation. *International Conference on Knowledge Engineering and the Semantic Web*. Springer, pages 242–249, 2013.

[14] J. D. Fernández, M. A. Martínez-Prieto, C. Gutiérrez, A. Polleres, and M. Arias. Binary rdf representation for publication and exchange (hdt). *Web Semantics: Science, Services and Agents on the World Wide Web*, 19, 2013.

[15] S. Franklin and A. Graesser. Is it an agent, or just a program?: a taxonomy for autonomous agents. *International Workshop on Agent Theories, Architectures, and Languages*. Springer, pages 21–35, 1996.

[16] W. Gödert. An ontology-based model for indexing and retrieval. *CoRR*, abs/1312.4425, 2013.

[17] W. Gödert. An ontology-based model for indexing and retrieval. *Journal of the Association for Information Science and Technology*, 67(3):594–609, 2016.

[18] S. Harris and A. Seaborne. Sparql 1.1 query language. Recommendation. World Wide Web Consortium, March 2013. `http://www.w3.org/TR/sparql11-query/`

[19] J. Hendler. Agents and the semantic web. *IEEE Intelligent systems*, 16(2):30–37, 2001.

[20] J. Hendler. Is there an intelligent agent in your future? *Nature*, 11, 1999.

[21] G. Klyne and J. J. Carrol. Resource Description Framework (RDF): concepts and abstract syntax. Recommendation. World Wide Web Consortium, February 2004. `http://www.w3.org/TR/rdf-concepts/`

[22] D. W. Loveland. On indefinite databases and the closed world assumption. *Lecture Notes in Computer Science*, 138(18)18:292–308, 1982.

[23] R. Reiter. On closed world data bases. In: *Logic and data bases*, pages 55–76. Springer, 1978.

[24] G. Rizzo and R. Troncy. Nerd: evaluating named entity recognition tools in the web of data. *(ISWC'11) Workshop on Web Scale Knowledge Extraction (WEKEX)*, 2011.

[25] M. Schmidt, M. Meier, and G. Lausen. Foundations of sparql query optimization. *Proceedings of the 13th International Conference on Database Theory*. ACM, pages 4–33, 2010.

[26] A. Schwarte, P. Haase, K. Hose, R. Schenkel, and M. Schmidt. Fedx: optimization techniques for federated query processing on linked data. *International Semantic Web Conference*. Springer, pages 601–616, 2011.

[27] M. Vander Sande, R. Verborgh, A. Dimou, P. Colpaert, and E. Mannens. Hypermedia - based discovery for source selection using low-cost linked data interfaces. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 12(3):79–110, 2016.

[28] R. Verborgh, O. Hartig, B. De Meester, G. Haesendonck, L. De Vocht, M. Vander Sande, R. Cyganiak, P. Colpaert, E. Mannens, and R. Van de Walle. Querying datasets on the web with high availability. *The Semantic Web–ISWC 2014*, pages 180–196, 2014.

[29] R. Verborgh, M. Vander Sande, O. Hartig, J. Van Herwegen, L. De Vocht, B. De Meester, G. Haesendonck, and P. Colpaert. Triple Pattern Fragments: a low-cost knowledge graph interface for the Web. eng. *Web Semantics*, 37-38:184–206, 2016.

[30] J. Waitelonis, M. Knuth, L. Wolf, J. Hercher, and H. Sack. The path is the destination–enabling a new search paradigm with linked data. *Proceedings of the Workshop on Linked Data in the Future Internet at the Future Internet Assembly*, 2010.

Part II

Techniques

Chapter 4

Interactive Search Visualization

If you feel that you're not getting enough out of a song,
change the instrument
- go from an acoustic to an electric or vice versa,
or try an open tuning. Do something to shake it up.

—Mark Knopfler.

This chapter discusses the exploration technique focusing on the front-end. Looking into front-end aspects addresses the different kinds of user activities of exploratory search introduced in Chapter 2. A lack of in-depth understanding of the inherent complexity of linked data graphs and the many degrees of freedom in modeling and querying of data limits many users to optimally query and interpret linked data. Therefore, this chapter explains an interactive visual graph-based workflow. It details how useful the workflow is for users to explore data and the relationships between the data. Furthermore, this chapter describes the architectural model, its implementation and illustrates the potential of interactive search visualization from a users point of view, both in terms of the workflow as well as the visualization. The majority of the evaluation respondents welcomed the workflow and considered its potential for linked data exploration and the insights they can get out of it.

4.1 Introduction

Interactive visual search goes further than the paradigm of keyword-based search or lookup based information retrieval, introduced in the beginning of Chapter 2. In keyword-based search a user would repeatedly try out different results and if not satisfied retry the search with slightly or completely different keywords. As linked data are typically represented as graphs [6], exploring their visualization as such is one of the ways to allow users to implicitly compose queries, identify links between resources, and intuitively discover new relevant pieces of information [3]. This chapter discusses a workflow that uses an interactive visualization to facilitate linked data query formulation. Background processes seek additional relations between the search results and present them as alternatives to the already delivered results. In this way, users are guided in expanding or narrowing down the range of facets available corresponding to a certain search query. This offers the users each iteration several exploration options and involves new and already found items in the search.

To this end, we considered for our workflow: (i) *exploratory data analysis* (EDA) [23] to assist data consumers to analyze the available dataset; and (ii) *exploratory search* [19] to facilitate them synthesizing complex queries.

EDA allows the data itself to reveal its underlying model and its relationships without requiring any formal statistical modeling and inference (non hypothesis-driven). Graphical EDA employs a variety of techniques to present the underlying data, maximizes the insight into a dataset and uncovers the underlying data patterns, allowing the users to discover the resources in the dataset. Exploratory search, on the other hand, describes either the problem context that motivates the search or the process by which the search is conducted. The users start from a vague but still goal-oriented defined information need and refine their need upon the availability of new information to address it, with a mix of keyword look-up, expanding or rearranging the search context, filtering and analysis.

The challenge we address here is the way a visualization involves the users and lead them through facets expressed in visually recognizable dimensions (e.g. shape, size, color etc.) rather than textual representations (e.g. lists). The key variables (KV) are broadly used (see the related work in Table 2.1). They address research questions **RQ2** and **RQ4**, test hypothesis HYP2 and give insight on the perceived usefulness of the added interactivity and visualization:

interactivity user perception of the visualization in terms of the goal;

effectiveness productivity of the way resources are shown to users;

features the impact of personalization, centering the search around the users based on their social media profile, and discovering links between resources shown to the users.

Other related work covers the creation of different views of linked data or study in particular use cases evaluating prototype interfaces. One editor that facilitates the process of creating web-based visualizations relying on linked data is "Visualbox" [16]. The developers concluded though that their editor was still too general for users to work practically with a visualization. However, test users valued that once a query was ready, the construction of a visualization was trivial. The "Linked Data Visualization Model" (LDVM) allows to connect different datasets, data extractions and visualizations in a dynamic way [1] rather than focusing on a single platform. Tvarozek et al. [24] empower users with access to semantic information spaces via an exploratory browser. At the end of an exploration session users need to start a new search, a history view allows users to step back. Dadzie and Rowe [3] concluded in their study on linked data interfaces and visualizations that only a limited number was available at the time of writing and each of them focuses on a separate aspect to support users. They highlighted the issue, an important motivation for our workflow as well, that without good quality linked data there is little motivation to build such interfaces for end-users while these interfaces are needed to locate and retrieve linked data in the first place.

4.2 Architectural Model

During interactive visualization of search results, users interact with the visualizations and their actions are translated and refined to more precise or broader queries iteratively. In this chapter, we explain workflow to interact with the results visualization. Figure 4.1 shows a schema of the different techniques working together during exploratory search. Further details on the dynamics on this process are explained in Chapter 5 and onwards.

The workflow consists of four phases and starts from a broad overview towards a detailed narrow view which serves as starting point for further exploration [6]. Figure 4.2 shows how users start with an overview of the dataset (Figure 4.2*a*) through which the users "dive" in more narrow perspectives (Figure 4.2*b*) by selecting a group to find out details and see the internal relations of the subdivisions (Figure 4.2*b*). A

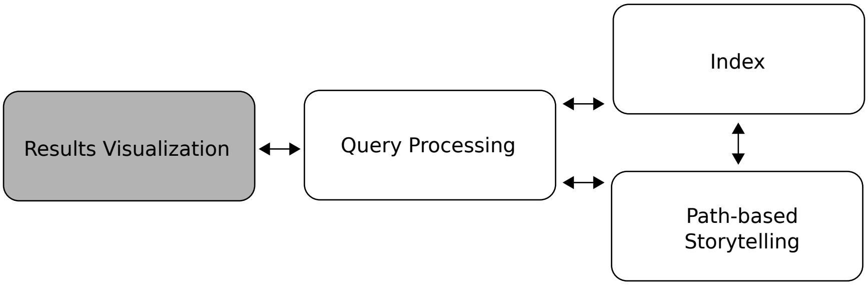


Figure 4.1: Interactive search results visualization relies on the combination of different techniques.

coordinated view (Figure 4.2*c*) of selected resources leads them through a broadened view (Figure 4.2*d*) by exploring relations of these resources.

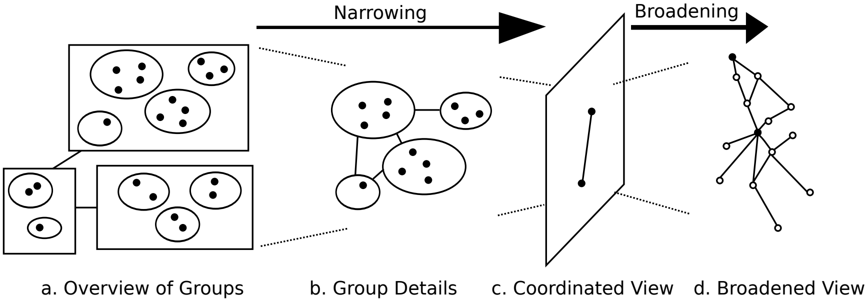


Figure 4.2: Narrowing views (*a, b*) allow users to analyse the dataset. The coordinated view (*c*) allows perspective switching in the workflow. Broadening views (*d*) allow users to explore the interlinked information beyond the dataset's boundaries.

Narrowing Views

The narrowing views (a, b) aim to familiarize the users with a certain dataset, as they are not aware of its context. The dataset itself reveals its underlying model and the relationships between its resources. Given the "unlimited" extent of a dataset, the initial view is focused on this certain dataset and its broader concepts are demonstrated. Exploration continues by following the links until reaching the resources that can not be further decomposed.

Coordinated View

The interactive visualization workflow is streamlined through a coordinated view [2] of the two different parts. This view centralizes the link focused on a specific resource that binds them. As the users, supported by the visualizations, narrow down to more detailed resources (a certain resource or the links between two resources), they reach the resources that cannot be further decomposed and thus act as the coordinated view (c). Starting from this view, the users, being aware of the underlying dataset, start exploring the dataset. The coordinated view forms a “bridge” between the narrow view and the broader view, which exploits existing links amongst resources across different datasets. The use of coordinated views to facilitate integration of visualizations [21], is a way to allow switching between visualization methods to successfully seek and discover information [20]. Coordinated views align multiple perspectives on a dataset.

Broadening Views

In the case of *broadening* views (d), data consumers find novel relations between existing and known to them resources interacting with visualizations of the data. The possible views are not limited to the data of the narrowing view but the links to other datasets are also revealed and visualized if considered relevant. It is a new way to search and explore the information. This way, users get an overview by using an approach that visualizes the search process interactively in, e.g. our aligned linked data knowledge base of related resources.

Applying Information Exploration Techniques to the Workflow

The narrowing view is achieved based on exploratory analysis techniques [23] applied to the dataset. Without any formal modeling or assumption about the underlying dataset, the main concepts and their relationships are gradually revealed. Subsequent views narrow the broader concepts and reveal more details about the relations among the concepts. The broadening view is achieved using exploratory search techniques [19] over individual instances of the linked data. Users iterate over individual concepts, their direct neighbors and their relationships. Iteratively expanding and focusing the visualization leads to more insight in selected concepts in the datasets. This way, the workflow enables users to discover, search and analyze linked data.

4.3 Implementation

A combination of two tools implements the interactive search visualization and workflow. The result is a graph based exploration interface supporting narrowing views by *LOD/VizSuite* and broadening views by *ResXplorer* over a coordinated view. The implementation uses researcher and academic library metadata as example, more details on the data used and the conducted experiment are given in Section 4.4.

LOD/VizSuite. The goal of the LOD Visualization Suite (LOD/VizSuite)[1] is to create an easily customizable visualization framework on top of LOD. LOD/ VizSuite aims to be data and schema agnostic, therefore it can be easily transferable to visualize different datasets. Its functionality is based on SPARQL queries which are published as SPARQL templates. Parameters can be passed to the SPARQL template at request time, which replace placeholders to construct a valid SPARQL query. The SPARQL templates are published at a DataTank[2] instance, a RESTful (Linked) Open Data management system which publishes data on the Web.

ResXplorer. ResXplorer[3] is used by researchers to find novel relationships between existing known items such as authors, publications, or conferences. Users interact with a visualization of resources [7] using an interface combining an optimized pathfinding algorithm [5] with Web 2.0 technologies (such as JQuery and Django). The result is a semantic search tool providing both a technical demonstration and a visualization that is applicable to many other applications beyond academic library metadata.

Making Search Decisions

The decision making process during search is supported by a real-time keyword disambiguation. This allows users to select the intended meaning from a drop down menu that appears below the search box. Presenting candidate query expansion terms in real-time, as users type their queries, can be useful during the early stages of the search [25].

Users can define and select their 'intended' search goal over several iterations. A combination of various resources is then presented to the users. In case they have

[1] `http://ewi.mmlab.be/academic`
[2] `http://thedatatank.com`
[3] `http://www.ResXplorer.org`

no idea which resource to investigate next, they get an overview of possible objects of interest (like points of interest on a street map).

Embedding Visualizations in the Workflow

In this section, the visualizations are embedded in the architectural model and implement the three types of views of our exploration workflow. Figure 4.3 shows the visualizations embedded in the workflow.

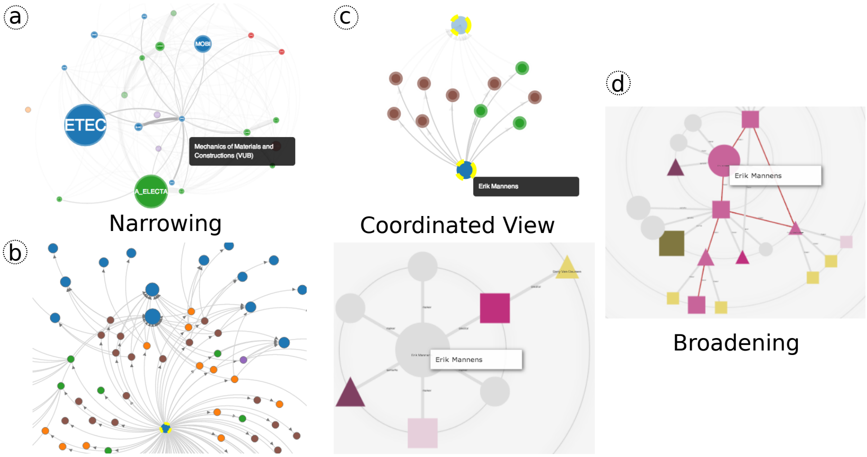


Figure 4.3: Corresponding with steps *a, b, c, d* in Figure 4.2, users narrow down from disciplines (*a*) to research groups and further to the individual researchers in this group (*b*). To find out relations between researchers they select two researchers and, using the coordinated view (*c*), shift to the broadening view and expand to resources beyond their research community (*d*).

Narrowing Views. The broadest concepts, which cover all the dataset, are chosen for the *overview view*. The *overview view* serves the users to discover the *main concepts* of the dataset, the *strength of the relations* between them and the *diversification of the total number* of the instances that constitute the broader concept. From the overview view, the users discover the narrower entities. *Broader views* are achieved by aggregating narrower entities using SPARQL queries that select and group them. Visualizations that provide an *overview view* of a topic or type are achieved by aggregating the underlying entities as groups and providing links considering of things they have in common. The groups are shown as graph nodes that diversify in size depending on the total number of common things of a certain topic or type they have, while the strength of the links depends on other commonalities of the entity.

Each group is the aggregation of individual entities, a user can further narrow down and view the entities and their commonalities (*decomposed views*). This is the narrowest view which acts as the *coordinated view*.

Coordinated View. In our use case such a resource can be a single entity or the links between two entities whose extensive commonalities are shown. As the end users view the network formed around a researcher or the exhaust list of paths between two researchers, they can be transposed to the corresponding view of the broadening part of the workflow.

Broadening Views. By the time the users explore the dataset, they can start expanding the network. While exploring the *broadening view*, data consumers are not limited to the data of the dataset but their exploration is enhanced with links to other datasets of the linked open data cloud that might be relevant to their exploration (e.g., DBLP in our use case).

4.4 Evaluation

Based on the implementation of the workflow for a use case concerning academic library metadata (which is further detailed in Chapter 8) we evaluated:

(i) **The Workflow**: how different aspects are perceived by the users: the usefulness, exportability, complexity, learnability, and innovation potential [interactivity];

(ii) **Embedding Visualizations in the Workflow**: assessed the end-user effectiveness, and productivity of the visualizations in the workflow [effectiveness];

(iii) **Feature Impact**: the impact of personalization and pathfinding as features in the visualization [features].

Methodology. Exploratory search represents a cognitively intensive activity. Therefore conduction of searches should be possible with minimal interruptions. According to White et al. [25, 26] : "Techniques such as questionnaires and interview techniques can be valuable tools, but one must be careful to include them in the experiment in such a way as to not interfere with their exploration". The choice of evaluation methodology was made by applying relevant aspects out of already existing achievements in this field introduced in [15, 17, 25] and adapting them to our specific use case. Since we want to offer a solution for research and learning purposes but also for wider community of users, a user centered methodology plays a decisive role in our evaluation process.

We evaluated the tools in two ways: end-user tests and expert user reviews. Both ways gave us insight in how the users perceived the tools and showed us potential bottlenecks [16]. They also delivered us insights on how precise our solution performs in comparison to the existing state of the art solutions of industry as well as academia. This evaluation includes a summary of the most important results explained in our work [8, 12], where we selected experts and researchers in computer science and digital media as test group representatives. We asked the test-users to participate in a controlled experiment - to find a relevant person to contact or a conference to attend.

Datasets. LOD/VizSuite provides visualizations based on the linked open data provided by the *"Research Information Linked Open Data"* (RILOD) data-set. RILOD is the result of the integration of heterogeneous sources related to research in Flanders, ending up in a rich and diverse dataset. The datasets contain resources of researchers from the region of Flanders, their publications and projects, which are associated with the corresponding research groups and institutes, and classified under the IWETO Discipline classification [4]. LOD/VizSuite exposes research and collaboration networks, communities of practice in a certain discipline [13] and timelines to monitor a discipline's evolution over time [14].

ResXplorer uses the *"Digital Bibliography and Library Project"*[5] (DBLP), an on-line reference for computer science bibliography for bibliographic information on major computer science publications [18]. The binding between RILOD and DBLP is their content's intersection: the same researchers and publications appear in both datasets.

Furthermore the computer science publications are aligned with data from call for papers from *"Conference Linked Data"* (COLINDA). COLINDA was added to the Linked Open Data Cloud in 2015 [22]. COLINDA exposes information about scientific events (conferences and workshops) for the period from 2002 up to 2015. Besides title, description and time COLINDA includes venue information of scientific events

[4] `https://www.ugent.be/en/research/research-staff/iweto/`

[5] `http://dblp.l3s.de/`

which is interlinked with Linked Data sets of GeoNames, and DBPedia. Additionally information about events is enhanced with links to corresponding proceedings from the computer science bibliography, DBLP (L3S). The main sources of COLINDA are WikiCfP and Eventseer. The research questions addressed by this work in particular were: how scientific events can be extracted and summarized from the Web, how to model them in Semantic Web to be useful for mining and adapting of research related social media content in particular micro blogs [11], and finally how they can be interlinked with other scientific information from the Linked Data Cloud to be used as base for explorative search for researchers.

The Workflow

For the evaluation of the workflow, data was gathered using a multi-method approach: *observing* their behavior while interacting with the tools, noting their actions and an end-user *survey*. We used the think-aloud protocol during the experiments to collect feedback from the participants and we recorded the screen actions of participants using QTrace [6]. Apart from the 17 users who participated in the evaluation, 19 additional users participated in the evaluation by filling out the same survey after receiving information about the visualizations, giving us richer data for the survey analysis. According to Faulkner [15], the number of test users is enough to reach nearly high level of certainty for finding the most of the existing usability problems. This way we gained a broader group of respondents, giving us richer data for the survey analysis.

We kept the audience of the assessment broad by conducting also semi-structured interviews with various stakeholders. All of them are likely to be affected by the impact and value of accessible and explorable linked data. The use case was situated in the context of research information. Thus interviewees are active for the Flemish government department of Research and Innovation, the Department of Research policy from Ghent University, and, from a commercial point of view, in the domain of Business Development & Academic Relations.

Observations. The think-aloud analysis gave us information regarding the perception of the visualisations by the participants, during the executions of the assignments. Via this direct feedback, we concluded that test users are able to reason via

[6] http://www.qasymphony.com/qtrace.html

the tools: for example by appointing missing research groups in the visualizations, or by putting the size of the nodes into discussion in LOD/VizSuite.

The observations give us further insights regarding how the users expect the exploration to happen: clicking on the broad views, e.g., clicking on research groups within disciplines, they expect they get an intermediate overview and each step forward in the workflow can give additional input to explore. Once they realize the fact of the narrowed view and the effect of the coordinated view, users are able to fully comprehend the workflow and start with simple reasoning that supports the intention of the exploratory search tool.

We observed that, once test users comprehend the exploration workflow, they better accept the visualized data and become more able to form their exploration path. This affects their exploration behavior: they use the different features to get further insights (search query's, top affinity suggestions, or expanding via node clicking). Although complexity raises within the visualizations during the explorations (earlier explored data stay visualized), test users understand the potential of visualizing academic data and can name how they are related to another researcher via conferences or publications from intermediate researchers. Test users declared that further input could bring in additional points of interests.

End-User Survey. To evaluate the exploration, we asked the test users and twenty extra respondents their impression of the views using a questionnaire. We have collected for the evaluation typical keyword queries that have been asked by the target group of the use case ($N = 36$ users) [12], both researchers and innovation policy makers - all in the field of information and communication science, against the system during the evaluation of "usefulness" based upon the Technology Acceptance Model (TAM) [4]. They judged their experience with search interface on a Likert-Scale with values (Strongly Disagree, Disagree, Undecided, Agree, Strongly Agree). The result of the evaluation can be seen in Table 4.1. According to users, the interface is meant primarily to serve as an exploration interface which makes our approach focused more on the user experience and less on classical search issues.

To determine the impact and quality of the workflow considering their use, we analyzed how the users explored and perceived the visualizations in the corresponding views. We especially measured the perceived usefulness and learnability and how the participants estimate the potential of the visualizations.

Answer	Score	Variance
1. **Explore**	**4.00**	2.00
2. Discover	3.89	1.65
3. Search	3.42	1.70
4. Analyse	3.05	1.72
5. Clarify	3.00	1.78
6. **Tell stories**	**2.35**	1.62

Table 4.1: Results of the short survey on *perceived usefulness*.

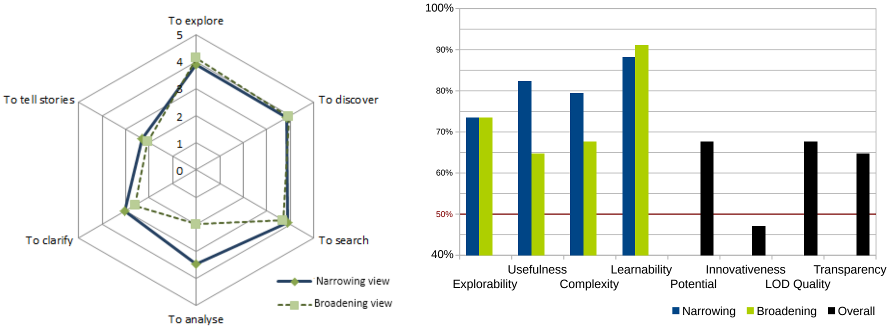


Figure 4.4: **User perceived goals of the views (left)** indicate that the narrowing view is perceived to be more suitable for analysis while the broadening view scores slightly better for exploration. **User satisfaction for the workflow (right)** shows that overall the views don't seem to expose innovation. In terms of usefulness and complexity the users are very satisfied with the narrowing, they need some time to learn how the broadening works.

Visual workflow's goals. To understand how the users perceive this visual workflow and its goals, we asked them to score possible purposes of use. As displayed in Figure 4.4, the respondents indeed perceived both the narrowing and broadening views as adequate tools to explore, discover and search. The broadening view is considered by the respondents as being a tool for exploration in the first place and discovery in the second place. The narrowing view is considered as a means to explore and to search.

Usefulness and Explorability. Test users agree that the visualizations are useful in terms of what it exposes 22 out of 34 (65%) agree for the broadening views and 28 out of 34 (82%) for the narrowing views. 28 out of the 34 respondents (74%) agree or strongly agree that the displayed relations of the broadening view are presented as an optimized selection of all results. Although, respondents stay rather undecided when it comes to the limitations: 16 out of 34 respondents (47%) agree or strongly agree that it is useful that the number of visualized resources and relations are limited, whereas 11 out of 34 (32%) disagree or strongly disagree on this. Finally, the respondents strongly agree that both the broadening view and the narrowing views support them gaining insights into the published data, but they were less confident in the case of the narrowing view.

Complexity and Learnability. The majority of the respondents agreed that they can learn quickly to interpret the visualizations both for the narrowing views, 27 out of 34 (79%), and the broadening views, 23 out of 34 (68%). Most of them think they found relevant insights at the narrowing and coordinated views as well during the broadening views. It is noteworthy that most of the respondents agree with the statement that once people get familiar with the visualizations in the narrowing view they can get benefit out of it, i.e. 30 out of 34 (88%) and even more of them agree for the broadening exploration, i.e. 31 out of 34 (91%).

Workflow Potential and Innovativeness. The respondents were asked how they perceive the potential of the workflow for linked data exploration. 23 of 34 test users (68%) state that the visual exploration workflow clearly helps them to understand the potential of Open Data. 16 out of 34 (47%) respondents agreed or strongly agreed that the visualizations help to get insight into innovation. However, 11 out of 34 (32%) respondents remained undecided.

Embedding Visualizations in the Workflows

In line with earlier work [12], we evaluated how appealing the workflow and the visualizations are to the end users by assessing the productivity and precision of the narrowing part and the complexity and searchability of the broadening view. Our evaluation showed that the implemented visualizations were capable of assisting the end-users to interpret the visualizations, thus adequate for the scope they were designed.

During a controlled experiment for evaluating the visualization aspect of the workflow, users were asked to think aloud and their actions were recorded while an evaluator observed the comments and took notes. Each test took about 30 to 45 minutes. We observed how the test users executed the assignment and we asked them to think aloud. The test users were asked (i) to start from their preferred research discipline (overview view), (ii) to go on towards their preferred research group and researchers and, explore their collaborations (explore the links of the narrower views) and (iii) to explore the links of one of the researchers that they concluded at while they navigated to broader views (broadening view).

Their assignment was as follows:

Assignment The users had to mark all found resources relevant to them. Then, users could choose between three actions: searching, adding top related resources; this is done through disambiguated keyword based search on topics knowingly related to the initial search term, e.g. choosing Tim Berners-Lee as initial keyword and WWW 2013 next related keyword in search, or expanding neighbors of found resources. In the last case they could chose between direct or indirect neighbors of the centrally focused node in the visualization. A 'top related' resource is the resource directly linked to the node in focus (centered) that shares the most common links with it.

Effectiveness measures how often a displayed result (R) related to a resource was marked relevant by the user (M).

$$\textbf{effectiveness} = E = \frac{\mid M \cap R \mid}{\mid R \mid} \tag{4.1}$$

Each action that delivered new resources to the result set resulted in an increase of quality of the result set.

Productivity measures this increase. The quality of a result set is the number of marked relevant resources compared to the total number of visualized resources. Productivity P_r measures the increase of effectiveness E_k after each test-user set of search actions in $A = \{a_1, ..., a_k, ...\}$:

$$\textbf{productivity} = P_r = \sum_{k \in A} \frac{E_k - E_{k-1}}{\mid A \mid} \tag{4.2}$$

where E_k is measured effectiveness after the action a_k. E_0 is the first measured effectiveness, so in the formula for P_r we note that $k > 0$.

The data in Table 4.2 shows that adding a top related resource was not done often by the users and added only a couple of resources to the result set. However, it proved

Table 4.2: Overview of user actions used to visualize new resources to measure the effectiveness of the actions

	Visualized (#)	**Marked Relevant (#)**
Search Resource	124	54
Add Top Related Resource	26	13
Expand Neighbours	94	34
Expand Neighbour of Neighbours	51	13
Expand Futher Related Resource	21	6

to be the most effective action as the users marked $^{13}/_{26}$ (50%) of the visualized resources relevant. The data in Table 4.2 also shows an increase of 12% in productivity in based on an average over all test users. This can be interpreted as follows: the search process is split into phases, where each phase is marked with a new set of visualized resources (thus changes in the result set), this figure indicated that when the result set changed, on average the new result set contains 12% more relevant resources than the previous view. For example if two new resources are added to a result set with 2/8 (25%) relevant resources and both are relevant, this results in 4/10 (40%) relevant resources, an increase of 15%. The average of all these increases is the productivity. Adding top related resources resulted in a result set that contained 12% more relevant nodes as before adding top related nodes.

As Table 4.3 indicates, searching for a resource was the most productive of all type of actions (+25%). This is remarkable as the user action effectiveness of searching is much lower than adding a top related resource on average over all test users. Adding top related resources resulted in a result set that contained +12% more relevant nodes as before adding top related nodes, even though it has higher effectiveness (50%). This means that the impact of each added resource when searching is much bigger, because the quality of the result set was not relatively high at the moment users decided searching.

Table 4.3: User action effectiveness and productivity (results on a scale from -100% tot 100%).

	Productivity (%)	**Effectiveness (%)**
Search Resource	25	31
Add Top Related Resource	12	50
Expand Neighbours	6	32
Expand Neighbour of Neighbours	1	25
Expand Futher Related Resource	1	29

The effectiveness of expanding resources $^{53}/_{166}$ (32%) is about the same as searching for a resource $^{54}/_{174}$ (31%). As the user actions resulted in about as many new resources in the case of searching and expanding, this is a very reliable comparison. Expanding the direct neighbours is the most productive (+6%) expansion. Expanding further related neighbours retains the quality of the result set and barely impacts it.

Feature Impact

We conducted a survey among the users to measure the impact of the two most important features of ResXplorer: *personalization* (using social media data, cfr. Chapter 7 for more details) and *paths* (enabling pathfinding). We presented the users screenshots of result sets in ResXplorer in A/B pairs to measure the impact of the personalization and relation discovery features, with one of the features enabled, both disabled and both enabled. They were asked to rate on a Likert scale from -3 to $+3$, from more towards A to more towards B, which result set they preferred for simple and complex queries. They did not know in advance which one had which feature(s) enabled. A simple query could be for example: 'Finding research publications that share (co-)authors with another paper (optionally: which the user also contributed to)'. Complex queries solve tasks like: 'Finding at least two people that presented a paper two years in a row in a certain conference series (optionally: where the user also had presented a paper)'.

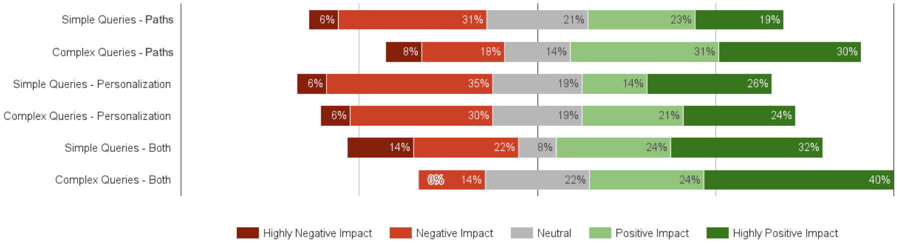


Figure 4.5: Impact on the result set relevancy of ResXplorer features according to users.

Figure 4.5 shows disagreement or no clear positive impact for simple queries when pathfinding is enabled and a rather negative impact when personalization is enabled for simple queries. A possible explanation is that for simple queries, the personalized results seemed to introduce extra overhead in the search results. The additional relationships to the user included in the results are not always that beneficial compared to the cases when the query was complicated. Many relationships were shown already in the case of complex queries, so there is probably less overhead. The results are more positive where more than 60% of the users agrees that for complex queries the results when using pathfinding are preferred. For personalization the ratio is 45% positive against 36% negative, the bias is less positive here, but clearly better than the case for personalization with simple queries. When looking at enabling both features vs. disabling both features, nearly 66% prefers the results with both personalization and pathfinding enabled and 56% in case of the simple queries.

Discussion

The results on **interactivity** learned that when users get familiar with the workflow: the narrowing and coordinated view helps them in the discovery and exploration of the linked data published in a dataset; and the broadening part helps them to discover, find new insights and explore the links of the data in the dataset with the data from the broader Linked Open Data cloud.

Analyzing the observations for the **effectiveness** and productivity indicated that searching by keywords for resources increases the result set with the most new relevant resources, while it is on average as effective as expanding existing resources in the result set. The most effective user action was adding top-related nodes to the visualization. The results on the impact of **features** are in line with previous studies we did on the dynamic alignment of social data with conference publication data [10] and the usability study of the "Researcher Affinity Browser" [9]. All these findings back the emphasis at several places in the paper on the positive influence of pathfinding and personalization in exploratory search.

The results of the productivity are affected by the technique for processing and translating the user actions to queries for the underlying index structure, this is the main topic of Chapter 5. The analysis of the contribution of the features for finding relationships between resources is subject of Chapter 6.

Summary

An interactive workflow for search based on visualizations allows users to have a unique, multifaceted experience when combined with techniques for information exploration. Two such interfaces were implemented to demonstrate the workflow for the exploration of an example dataset. Even though the workflow forces users to interact with the data in a certain way, different and unfamiliar to them, at the end it achieves its goal and users become acquainted with the underlying dataset and can bring in new unexplored information and knowledge. According to the respondents, they, as users, became acquainted with the underlying data and found the workflow to bring in new unexplored information as soon they familiarized themselves with the workflow. According to the users the visualization pinpoints resources for researchers efficiently and effectively. Considering that the implementation of the visualizations are still in the prototype phase, the potential of a visual and interactive search interface is well demonstrated and understood by the users.

References

[1] J. M. Brunetti, S. Auer, and R. GarcıÍa. The linked data visualization model. *International Semantic Web Conference (Posters & Demos)*, 2012.

[2] C. Collins and S. Carpendale. VisLink: revealing relationships amongst visualizations. *IEEE Trans. on Visualization and Computer Graphics (Proc. of the IEEE Conf. on Information Visualization (InfoVis))*, 13(6), 2007.

[3] A.-S. Dadzie and M. Rowe. Approaches to visualising Linked Data: a survey. *Semant. web*, 2(2): 89–124, April 2011.

[4] F. Davis. A Technology Acceptance Model for Empirically Testing New End-user Information Systems: Theory and Results. Massachusetts Institute of Technology, 1985.

[5] L. De Vocht, S. Coppens, R. Verborgh, M. Vander Sande, E. Mannens, and R. Van de Walle. Discovering meaningful connections between resources in the web of data. *Proceedings of the 6th Workshop on Linked Data on the Web (LDOW2013)*, Rio de Janeiro, Brazil, 2013.

[6] L. De Vocht, A. Dimou, J. Breuer, M. Van Compernolle, R. Verborgh, E. Mannens, P. Mechant, and R. Van de Walle. A visual exploration workflow as enabler for the exploitation of linked open data. *Proceedings of the 3rd International Workshop on Intelligent Exploration of Semantic Data*, 2014.

[7] L. De Vocht, E. Mannens, R. Van de Walle, S. Softic, and M. Ebner. A search interface for researchers to explore affinities in a Linked Data knowledge base. *Proceedings of the 12th International Semantic Web Conference Posters & Demonstrations Track*. CEUR-WS, pages 21–24, 2013.

[8] L. De Vocht, S. Softic, A. Dimou, R. Verborgh, E. Mannens, M. Ebner, and R. Van de Walle. Visualizing collaborations and online social interactions at scientific conferences for scholarly networking. *Proceedings of the Workshop on Semantics, Analytics, Visualisation: Enhancing Scholarly Data (SAVE-SD 15); 24th International World Wide Web Conference*, 2015.

[9] L. De Vocht, S. Softic, M. Ebner, and H. Mühlburger. Semantically driven social data aggregation interfaces for research 2.0. *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*. i-KNOW 2011, pages 43:1–43:9, ACM, Graz, Austria, 2011.

[10] L. De Vocht, S. Softic, E. Mannens, M. Ebner, and R. Van de Walle. Aligning web collaboration tools with research data for scholars. *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*. WWW Companion 2014, pages 1203–1208, International World Wide Web Conferences Steering Committee, Seoul, Korea, 2014.

[11] L. De Vocht, D. Van Deursen, E. Mannens, and R. Van de Walle. A semantic approach to cross-disciplinary research collaboration. *Internation Journal of Emerging Technologies in Learning (iJET)*, 7(S2):22–30, 2012.

[12] A. Dimou, L. De Vocht, M. Van Compernolle, E. Mannens, P. Mechant, and R. Van de Walle. A Visual Workflow to Explore the Web of Data for Scholars. *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*. WWW Companion '14, pages 1171–1176, International World Wide Web Conferences Steering Committee, Seoul, Korea, 2014.

[13] A. Dimou, L. De Vocht, G. Van Grootel, L. Van Campe, J. Latour, E. Mannens, P. Mechant, and R. Van de Walle. Visualizing the information of a Linked Open Data enabled Research Information System. *Proceedings of the Current Research Information Systems Conference*, 2014.

[14] A. Dimou, L. De Vocht, R. Verborgh, E. Mannens, and R. Van de Walle. Visualizing research networks' evolution over time. *Proceedings of the Workshop on Intelligent Exploration of Semantic Data (IESD) at the International Semantic Web Conference (ISWC).* 2016.

[15] L. Faulkner. Beyond the Five-User assumption: benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments, & Computers*, 35(3), 2003.

[16] A. Graves. Creation of visualizations based on Linked Data. *Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics*. ACM, pages 41, 2013.

[17] W. Kraaij and W. Post. Task based evaluation of exploratory search systems. *SIGIR 2006 workshop, Evaluating Exploratory Search Systems*, 2006.

[18] M. Ley. The DBLP computer science bibliography: evolution, research issues, perspectives. *String Processing and Information Retrieval*. Springer, pages 1–10, 2002.

[19] G. Marchionini. Exploratory search: from finding to understanding. *Commun. ACM*, 49(4):41–46, April 2006.

[20] J. C. Roberts. State of the art: coordinated & multiple views in exploratory visualization. *Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV'07. Fifth International Conference on*. IEEE, 2007.

[21] M. Scherr. Multiple and coordinated views in information visualization. *Trends in Information Visualization*, 2008.

[22] S. Softic, L. De Vocht, E. Mannens, M. Ebner, and R. Van de Walle. COLINDA: modeling, representing and using scientific events in the web of data. *Proceedings of the 4th International Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2015) Co-located with the 12th Extended Semantic Web Conference (ESWC 2015), Protoroz, Slovenia, May 31, 2015.* pages 12–23, 2015.

[23] J. Tukey. Exploratory Data Analysis. Addison-Wesley series in behavioral sciences. Addison-Wesley Publishing Company, 1977.

[24] M. Tvarožek et al. Exploratory search in the adaptive social semantic web. *Information Sciences and Technologies Bulletin of the ACM Slovakia*, 3(1), 2011.

[25] R. W. White and G. Marchionini. Examining the effectiveness of real-time query expansion. *Information Processing & Management*, 43(3):685–704, 2007.

[26] R. W. White, G. Muresan, and G. Marchionini. Evaluating exploratory search systems. *Proceedings of the ACM SIGIR 2006 Workshop on Evaluating Exploratory Search Systems*, pages 1–2, 2006.

Chapter 5

Query Processing

The question of whether a computer can think
is no more interesting than
the question of whether a submarine can swim.

—Edsger W. Dijkstra.

Search queries to find relevant content on the Web, typically consist of keywords that can only be matched in the content or its metadata. The Web of Data extends this functionality by bringing structure and giving well-defined meaning to the content and it enables humans and machines to work together using controlled vocabularies. Due to the high degree of mismatches between the structure of the content and the vocabularies in different sources, searching over multiple heterogeneous repositories of structured data is considered challenging. This chapter explains the efficiency and effectiveness trade-offs in for a query processing technique forming the bridge between the content from a user's perspective and its representation as machine-readable data.

5.1 Introduction

The way real world objects are shown to the user, be it visualized or described textually, differs from how they are represented in the back-end for machines, as raw data to be processed by algorithms. Because of this, translating what the users intend through their actions is not trivial. Some kind of bridge component responsible for interpreting the user intents and translating them to machine understandable queries is necessary. Because of this, every search action ends-up in this bridge component. The way the query processing interprets this action has an impact on the overall search efficiency (performance, complexity) and effectiveness (search precision). The metrics address research questions **RQ1** and **RQ4**, test hypothesis HYP1 and HYP3.

For example, a user wants to explore how two scientists, *Carl Linnaeus* and *Charles Darwin*, are related to each other, the user is able to recognize both as scientists by reading their names, by seeing a picture or reading a description. This is also the case when a user types in the names of the scientists in a search system, the user expects the system to recognize them as such. Without an intermediary step the string corresponding to the user input of the scientist's name, such as "carl linnaeus", would be fired directly upon the data representation in the back-end. The only possible way of linking would be to match the characters in the string with how in the contents - regardless of their data structure and aside from advanced algorithms that optimize the matching of text in the content (e.g. dealing with typos). There would be no way to be sure the documents containing the string input would give back the scientist the user is looking for. One of the things the bridge component takes care of is this lookup. Every entity, such as a person, is indexed with a type and a unique identifier. During the lookup of "carl linnaeus" the bridge component will retrieve all entities from the index matching the user's search with attributes such as the type and add a reference to the unique identifier - invisible to the user - under the form of a URI, in this case: *http://dbpedia.org/resource/Carl_Linnaeus*. This process appears to the user as a common typeahead or autocomplete functionality in a search input field. However, as the input of the user, is literally being hooked to an entity with a specific URI, it will allow the search to take into account semantics and to be more precise.

The evaluation section 5.4 shows the cases where and to which degree the query processing technique is more precise. The query processing technique described in this chapter acts as the internals of the bridge component between the user interface and the back-end. It facilitates reuse, and exposure linked data by coupling the results visualization and path-based storytelling that offer data or provide advanced algorithms on the data. It uses the 'affinity' of entities in relation to the initially formulated query as a measure for ranking the results. The main contribution of this technique is the way it adds an exploration aspect to interactive (visual) search and how it forms a bridge between the front-end interface and the back-end. This chapter continues with a broad description of the dynamics of this technique and details about its implementation before deep-diving in the the evaluation of its efficiency effectiveness. The top level interface, the user interface, delivers an aggregated and enriched view to users. All exposed content follows a common pattern from an aligned model through a protocol layer resulting in a semantically interpreted repository.

5.2 Architectural Model

Figure 5.1 depicts the overall architecture with the difference points of interaction between the different techniques. In this chapter we make abstraction of the Path-based storytelling technique which chapter 6 describes in detail. The query processing technique introduces three distinct modules that form the basis for preparing the data for exploration tasks and supporting end-users in exploring the data: *lookup*, *relate* and *rank*.

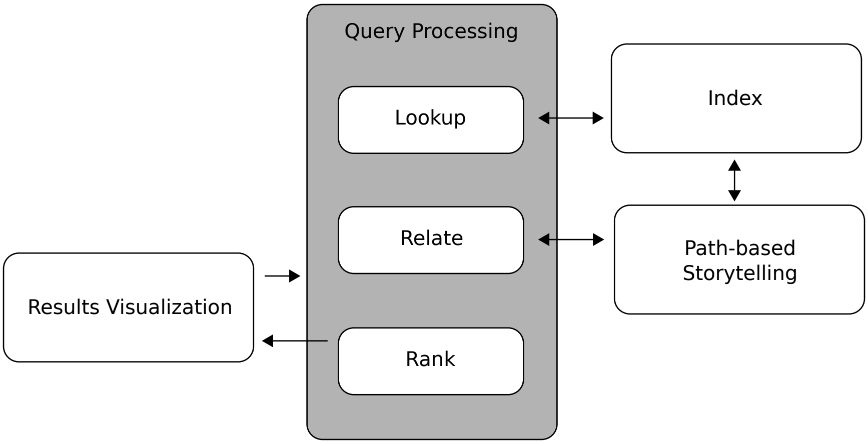


Figure 5.1: Keyword mapping, query translation and ranking within the search process through the query processing technique.

In this particular case we are interested in the effectiveness of the following modules

- *Lookup*, the translation of a keyword to a resource in a semantic representation, thus taking into account the unique reference, URI.
- *Relate*, functionality which looks into how pairs of resources are associated.

- *Rank*, functionality comprises the computation of a score of each of the resources in a particular search context. The rank can be used to order or visually tweak the appearance of the search results in the interface.

All modules of the query processing technique translate queries and result from one format to the other: they take input from users (keywords, resources or pairs of resources) and transform it to queries for the index or the path-based storytelling technique. The ranking module adds ordering with regards to the search context to it.

Looking up Resources

The module responsible for looking up resources can have two types of input: keywords or resources. In the case of keywords, they are forwarded directly to the search index. The index will return one or more resources, depending on its configuration of the index. Resources are usually typed with the *rdf:type* property. This property can be used to assign a category to each resource. In the case of a resource, the index will again be requested to provide more information about the resource requested. This can be detailed information such as a label, description or other resources it points to. In both cases the results are converted into a format that a component taking care of search results visualization understands. The look-up of the resources facilitates high-precision interactive search, because it succeeds in mapping annotated and interlinked structured data with ontologies from the various indexed repositories in an effective way. Figure 5.2 shows an example of a user iteratively looking up the terms *Linked Data*, *WWW 2012*, *Germany* which the lookup module retrieves from the index which returns the results. In the example the matching of keywords with resources is relatively loose: case insensitive, space insensitive, and search takes place over multiple indexed fields (tag, label, keywords). Depending on the configuration of each index in this regard, results may vary.

Once the user retrieved matches for the keywords, the user can choose to expand one or more of them further. In the example shown in Figure 5.3 the user chooses to find out more about the resource related to *Germany*.

Relating Resources

Each iteration, typically lookup-actions, results in more and more resources in a user's search session. Here the module that relates resources has a crucial rule. Be it

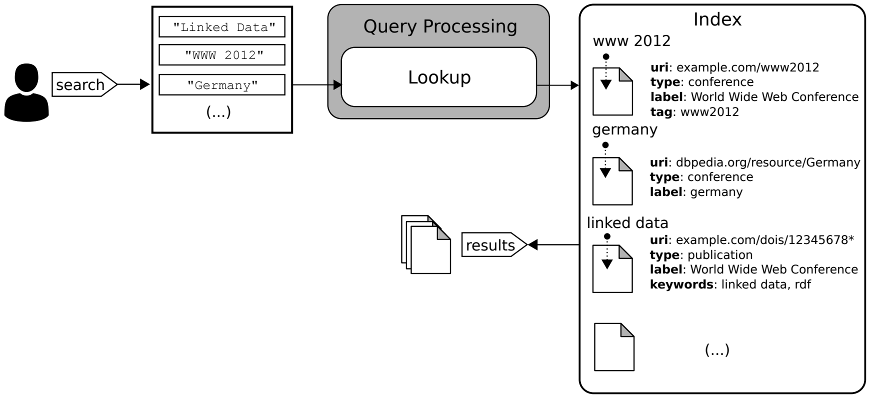


Figure 5.2: Looking-up resources by keyword.

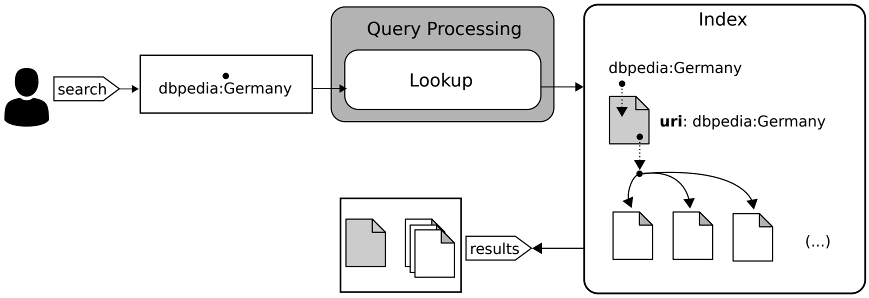


Figure 5.3: Looking-up details about a certain resource.

the user explicitly asking to relate a pair of resources or an automated request based on the resources currently shown in the results to the user, the query processing technique takes in the resources that need to be related and transforms it to queries for the path-based storytelling component. Figure 5.4 shows an example query where the user is interested in relating the resources bound to *France* and *Germany*. Again, just like when looking-up resources by keyword or retrieving more details about a resource, the results are being translated to a form that can be interpreted by the search results visualization.

Ranking Resources

Ranking resources and relationships in the Web of Data differs from traditional document ranking because semantic search engines interpret query results and their relation to the data sources. In traditional, document-based ranking, the ranking

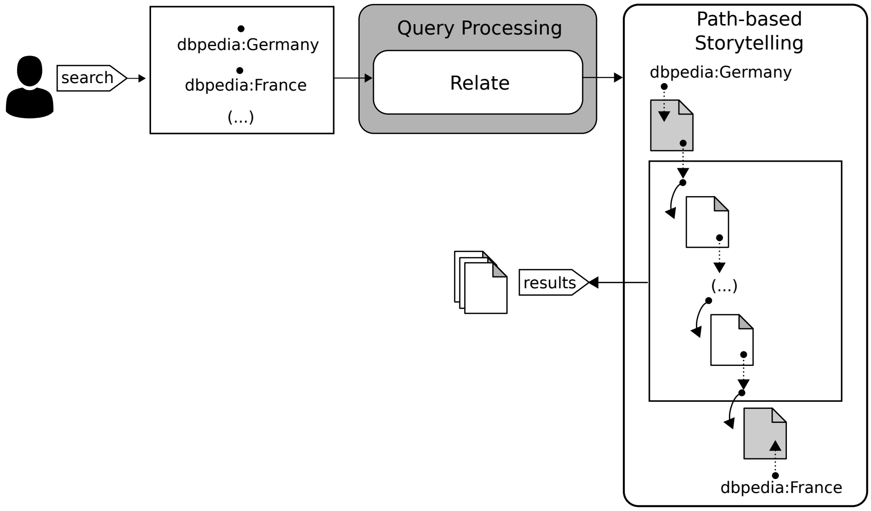


Figure 5.4: Relating two (or more) resources.

boils down to (re)-ordering the resources and optionally giving them a certain score. This is shown conceptually in Figure 5.5. Aleman-Meza et al. have demonstrated

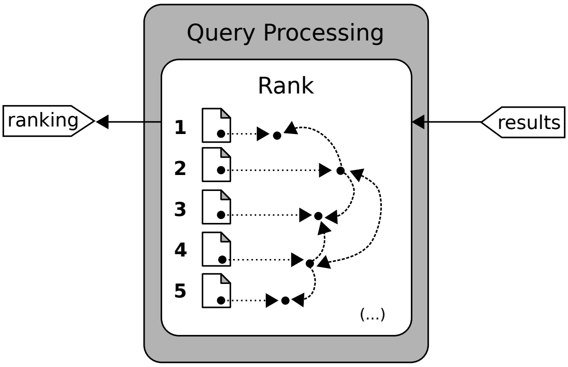


Figure 5.5: Ranking multiple resources.

the effectiveness of a flexible ranking approach that distinguishes between statistical and semantic metrics [1]. They used proximity to the search context as an important metric. Because it is critical for the success of an interactive tool for research [12], a ranking should take into account the discovery of newer unexpected relations. This has been applied in SemRank, which is a method for top-k ranking of semantic relations in search results [2]. Their approach permits changing a parameter to switch between a targeted search and a pure discovery mode. In pure discovery mode higher rank values are being assigned to the most unpredictable paths. Daoud et al. have shown the effectiveness of a personalized graph-based ranking model [4]. By considering cross links between graphs and distances between nodes, the work described in this chapter achieves personalization by affecting the original ranking of resources. Pintado et al. identified relationships, using dynamical and statistical analysis, between classes and objects and used metrics to quantify these relationships in order to express them in terms of object affinity in the context of Software Engineering [16]. Their goal and interface is similar to ours and the introduced concept 'affinities' is characterized by high levels sharing of similar properties and relations. Therefore, we apply this concept as a base for defining our ranking approach. The difference in the way we apply the concept affinity is in the interpretation and visualization. In Pintado, the affinity represents an object similarity measure which is computed based on class hierarchy, inheritance, composition etc. It can be seen as a projected multi-dimensional vector space with multiple possible slices. In our interpretation, the affinity is computed based on factors involving the object 'resource' semantics, its presentation will always include the graph topology and show resources that are in the graph linked to each other. The presentation is not based on a vector space projection. The visual positioning does not depend on some kind of object or semantic similarity measure. However, the affinity results in a number that the expresses the 'rank' which may be used to express the: order of which resources are shown, a threshold for a resource to be shown, or the size of the resource.

Pre-Ranking. Before we rank the relations between resources, the candidate resources to be included in relations are pre-ranked. The pre-ranking takes "popularity" and "rarity" into account, essential components in the PageRank algorithm [15] and is used to sort candidate related nodes in the proposed engine. The implementation takes these relations into account by using the Jaccard-coefficient to measure the dissimilarity and to assign a random-walk based weight, which ranks more rare resources higher, thereby guaranteeing that paths between resources prefer specific relations over general ones [13].

Affinity Ranking. We identified three important criteria for ranking in a search engine according to the architectural model:

C_R *proximity* to the search context;

N_R *novelty*, the discovery of newer unexpected relations to exceed predictable fact retrieval;

P_R *personalization.*

Alternatively, we quantify the relationships to help researchers focus. These metrics are always executed between an object pair. The path between them represents whether they are directly connected or not. The results are limited and optimized according this ranking mechanism.

The remainder of this section gives an overview of important semantic ranking criteria and explain why they are useful for our affinity ranking approach and discuss how they contribute to measuring affinity for a resource A_R. We define this hybrid ranking criterion as:

$$A_R = w_c * C_R + w_n * N_R + w_p * P_R \tag{5.1}$$

where we make sure that the weights are normalized to an application global configured constant k:

$$k = w_c + w_n + w_p \tag{5.2}$$

For each criterion users can configure a weight w, this can be used to optimize the focus on resources. In our evaluation we show the effectiveness of our search infrastructure with the presented ranking criterion and make a distinction between a personalized ($w_p = w_c$) and anonymous search case ($w_p = 0$). Proximity to the search context C_R is one of the main indicators of affinity. Novelty N_R and personalization P_R then refine the ranking further. It is very important that the weights in the affinity criterion A_R are adequately configured. Depending on the use case different factors might be more important than others. Novelty becomes more important when differences in type of relations are essential, so w_n should be relatively high. The amount of personalization can be taken into account as well by making w_p greater than 0, typically in the order of magnitude of w_c. All weights are relative to the proximity, which always is taken into account ($w_c > 0$). The weights depend on the application and the goal of the use case.

Proximity. In our case, the proximity to the context marks the number of relations found in a path between two resources, that belong to the search context. The context can be initiated by a user profile if the user so desires. Found resources can be related to it to personalize the ranking. In an anonymous search, the relationships binding the resources that represent the researchers input query keywords determine the context. We measure "proximity" - *how semantically related resources are*. A set of objects that are close in one context can seem quite unrelated in another context. Distance between the resources (path length) is another way of looking at this ranking criterion for the context. The further the distance between two resources is, the less related they are, since the increasing distance between the two resources also brings with it the fact that they do not really relate to each other, but have common intermediate resources which relate to them both. This on its own however does not guarantee a high quality relation between the two resources at the start and end of the path.

After we have defined the resources and relations belonging to the context C we define for each other resource R, out of the context, the proximity criterion C_R such that

$$d = distance(C,R) = \min_{C_k \in C} distance(C_k,R) \tag{5.3}$$

d is the minimum of $distance(C_k,R)$, the distances of the optimal cost paths between each resource C_k in the context C and R, as computed by the search engine. The optimal cost path depends on the path algorithm's configuration. For the minimum distance $k = \text{min}$. We use d to normalize the expression and then look for each intermediate resource I_i in the path between C and R whether it belongs to the context or not. The path between C_{min} and R can be noted as: $(C_{\text{min}},...,I_i,...,R)$.

$$x_i = \begin{cases} 1, & : I_i \in C \\ 0, & : I_i \notin C \end{cases} \tag{5.4}$$

$$C_R = \frac{\sum_{i=0}^{d-1} x_i}{d} \tag{5.5}$$

The $distance(C,R)$ is at least 1. The context typically consists of the mapped keywords, the relations between those resources and their properties.

Novelty. In research, unexpected discoveries make interacting with the search results more interesting. Affinity with resources in research is greatly affected by new discoveries and always searching within the same kind of resources and relationships does not guarantee it. We want to encourage sudden shifts of paradigm in paths. More shifts lead to higher novelty. This means that if a path switches from relations that describe people to relations that describe countries, the novelty score will be high, depending on how different the new paradigm is from the original and how many shifts there are.

We compute novelty N_R for a resource R along the relations belonging to the path from the nearest resource c of the search context C. We need to define the domain D_i for a relationship R_i, typically these are all other predicates for which there exists a connection to, such that

$$n_i = \begin{cases} 1, & : R_i \notin D_{i-1} \\ 0, & : R_i \in D_{i-1} \end{cases} \tag{5.6}$$

which means that we check whether R_i belongs to the domain of the previous relation R_{i-1} and $i > 1$. D_0 is the domain of R_0 (the first relation in the path). Except for the first relationship we can thus compute the novelty of the relation belonging to the path between C and R.

$$N_R = \frac{\sum_{i=1}^{d-1} n_i}{d-1} \tag{5.7}$$

We note that $N_R = 1$ if none of the relations in the domain of the previous relation and $N_R = 0$ if all relations belong to the same domain.

Personalization. To optimize and ensure a personal ranking we need to properly connect the found resources with the user's profile. We combine the graph of resources and the graph of the user profile through common concepts and cross links connecting the two graphs. Even in an anonymous search session we can optimize the ranking of the found results according to the users search context defined by the input keywords and selected resource representations.

We define n as a property of the user. Each user has a set of properties U and an element of this set is n. We compute the personalization criterion P_R for a resource R as the averaged sum of all properties of R related to the personalized context, which consists of the properties n, resulting in following equations:

$$d_n = distance(R, n) \tag{5.8}$$

where the distance d_n between R and n is computed along the path between R and n. The exact configuration of the distance depends on the search engine and more specifically, the configured path algorithm. The inverse distance $\frac{1}{d_n} = 0$ if there is no path. We compute P_R by iterating over each $n \in U$.

$$P_R = \frac{1}{|U|} \sum_{n \in U} \frac{1}{d_n} \tag{5.9}$$

5.3 Implementation

The query processing technique leverages annotated semantic graphs by relying on the fact that the vocabularies used in them can be used to link the source repositories. The relate module expects that there is some other component, in this case the path-based storytelling component, that can relate a particular resource with more information about it. This could be resources directly connected tot the resource, but also indirectly connected resources. Similar data of different sources can thus be described in using the same terms, making it possible to explore these sources with the same queries. Each of these resources is connected through a link, and these links are semantically annotated. These annotations are important to rank the resulting resources but also to give an indication to the user of the meaning of each relationship.

Underlying datamodel

Semantically annotated data in RDF, supports a flexible mediator-exchange model. RDF has several very appealing properties that position it as the exchange model of choice. The implementation uses the RDF data model to exchange the data and the results because it can act as a flexible mediator between various applications and across diverse infrastructures of complex heterogeneous data. RDF provides a graph representation of data and frees the data modeller and application developer from a priori having to define a schema.

Deployment

Figure 5.6 shows the deployment of the implementation. The actual search 'engine' is spread over both client and server. As bridge between front-end and back-end it deals both with processing results for the user and taking care of more low-level dealing with data in the back-end.

On client-side the engine's main function is to act as controller, in model-view-controller software-architectural terms. It is there that the modules are located responsible for transforming the results for viewing and visualization, looking up resources and relations between resources.

On server-side it provides the business logic and model level abstraction. The main modules there are the search and provider function.

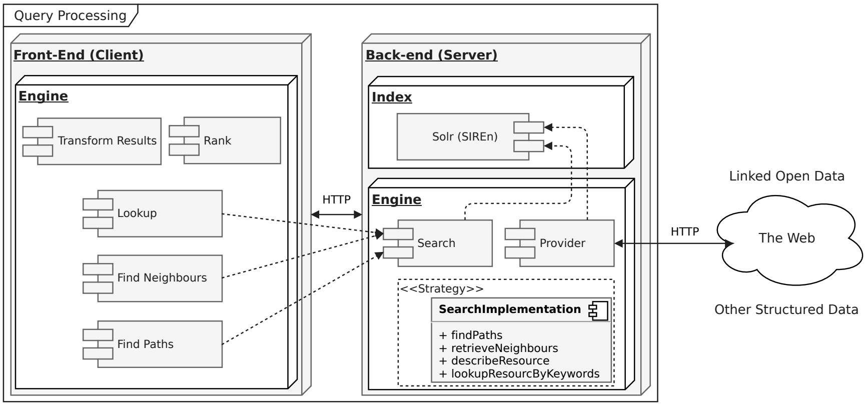


Figure 5.6: The query processing implementation is deployed both client-side and server-side.

The search function may be implemented depending on the strategy according to the 'strategy design pattern' [10]. The strategy design pattern allows the configuration of different contexts with the same programming interface and input-output format. The calls originate in this case from the user and going via the client-side part of the engine. Depending on the context, different strategies are chosen to answer to the call, but the response then again goes through the same interface. In a broad sense, the top-level modules consist of such contexts. Contexts for dealing with keyword lookup requests and contexts for dealing with resource requests, for example to find neighbours of a resource or paths between resources.
The provider function takes care of pre-loading the data from the Web into the index. It organizes data adequately for each resource from the configured data sources.

Index

Separate fields are foreseen for the unique identifier, type, label, and description. The remainder of the structured data is put in a separate field together without special distinction of certain attributes or relations. The properties *type* and *label* are indexed separately, because they are required for each linked data entity described in RDF[1] and allow retrieving entities by label and disambiguating them by type. The indices contain a special type of field *ntriple* that makes use of the SIREn Lucene/Solr plugin that allows executing star-shaped queries on the resulting linked data [7]. Star-shaped queries are essential to immediately find neighbouring

[1] http://www.w3.org/2009/12/rdf-ws/papers/ws17

entities for each entity and to ultimately find paths between non adjacent nodes. We chose Lucene because it is the defacto industry search engine for text search, it is stable, as its implemented based on state machines. Solr is implementation on of the most advanced HTTP layers on top of Lucene. Furthermore Solr was the only search framework at the time where a plug-in (Siren) was available that had a fast compressed data structure for storing triples inside a Lucene document. Even at the time of writing, Lucene/Solr and the more and more common ElasticSearch is still one of the most advanced search indices available.

5.4 Evaluation

We have evaluated the efficiency and effectiveness of a proposed query processing technique with a sample configuration and context. The execution of consecutive benchmarks facilitated tweaking the configuration for optimal back-end performance. Additionally we tested the retrieval quality with the sample test queries shown in Table 5.1. Expert users reviewed its information retrieval potential. The expert users did not interact through a user interface but were given the list of keywords and resources, literally as in the table, and the results were presented similar to the output of the query processing before visualization. They did not come in contact with, the more complex, underlying linked data model. They received the results under the form of ordered lists, according to the ranking scores, with dereferenceable URL's.

Another group of users which we must bear in mind are domain experts, as they are likely to have a very good understanding of data structure and content in their domain, and bring this knowledge to guide both browsing research and targeted searches. For this group we had to extend the sample to a full-fledged use case that included a proper interactive search interface as well. This evaluation is detailed in Chapter 8.

This section firstly explains a new benchmark model we used to measure the interaction between users, the semantic search engine and the interface. It then provides information about the datasets and finally reports on the applied and executed benchmark results for the experimental setup we implemented for our use case.

Benchmark Model

Existing work on benchmarks for semantic search, SPARQL queries, and linked data retrieval cover only the "bottom layer", the machine interface, of our needs for evaluation since our semantic search relies on user pre-sets and content the user published on social media. Some techniques of our model, like the path-based storytelling, use SPARQL queries for certain operations. Therefore, we considered the use of *SP2Bench* [18] and others alike, but it would not cover all aspects of the search functionality we implemented.

The efforts on defining benchmarks for semantic search are evolving [3, 9] and they delivered only single-experience recommendations so far. In the experience report [19] the authors: reflected about their experience over years on evaluation of semantic search systems (i); concluded that such evaluations are generally small scale due to the lack of appropriate resources and test collections, agreed performance criteria and independent judgment of performance (ii); and proposed for future evaluation work: "the development of extensible evaluation benchmarks and the use of logging parameters for evaluating individual components of search systems" (iii). Led by these findings and absence of adequate benchmarks that cover all facets of our approach we necessitated to define our own user-centered benchmark for social semantic search.

The goal of the benchmark is to evaluate the search engine with datasets relevant to researchers available on the Web of Data. The benchmark aggregates the other related approaches and optimizes aspects for use with interactive exploration, social data, Linked Open Data and involvement of end-users. As the benchmark focuses on end-users, the benchmark requires input from users to define the queries and measure the parameters.

Parameter Definition. The benchmark consists of variable parameters for input: a set of test queries Q and an experimental setup $X = \langle O, V, I \rangle$. X contains the semantic search engine under test O, an interactive search interface V and indexes static datasets S and a dynamic dataset, for example containing links to social media, D, in a search index I, so $I = \langle S, D \rangle$.

Baseline. As a baseline for the query engine O we used SPARQL transitive paths, basically giving the shortest possible chain of links connecting two entities. This is not standard SPARQL but is supported by some RDF stores like Virtuoso.

Query Selection. A set of n queries $Q = \{q_1, ..., q_n\}$ are identified by observing queries asked by at least N test-users in a controlled experiment, to guarantee a 'varied' mix consisting of distinct query patterns. Each query q_i consists of a number of keywords n_i fitted by selecting examples w_{k_i} in the query patterns the test-users were interested in: $q_i = \{w_1, ..., w_{n_i}\}$.

Indexed Datasets. It is important that both indexed datasets in the index I, the static S and the dynamic D, have sufficient links between them. If all test-users can start a personalized search, they can find out how several of their preferred keywords are related to their user profile. Each test-user profile is expressed as a set of triples in S.

Measures. The main parameters under test are the engine O and the interface V. The test-user interacts with the data through the interface V and the engine O is the bridging component between V and the datasets in the index I. All intermediary interfaces are optimized according to the semantic model for the selected datasets.

Key-variables.

We measure the *efficiency* and *effectiveness* to obtain insight in how well the system performs and its individual components interact. Each of these measures indicate a different aspect of the search engine.

Efficiency. The efficiency learns how the engine with its implementation E behave when parsing queries, such as the test set Q. The efficiency is divided in three independent sub-measures: (i) *quality*, (ii) *complexity*, and (iii) *performance*. The quality indicates how much relations between concepts after translating the keyword queries can be found. Complexity and performance focus on time and space (memory-usage) requirements for executing the translation and finding these relations.

Effectiveness. Effectiveness E on the other hand indicates the overall perception of the results by the users taking into account expert-user feedback. This is expressed as the search precision P [17].

$$P = \frac{\text{\# relevant objects}}{\text{\# retrieved objects}} \tag{5.10}$$

The reason why we have only measured precision but not recall is because computing relevant results for the entire dataset is complex due to its size and dynamic nature (D). However we can compute the relevance for each result set. Each query $q_i \in Q$ delivers a different number of relevant results, which makes the usage of mean average precision MAP an important measure. The aim of this averaging technique is to summarize the effectiveness of a specific ranking algorithm over the collection

#	Keywords
Q_1	LDOW, Bizer
Q_2	ISWC2012, Lyon, France,
Q_3	ISWC2008, linked data, Germany
Q_4	linked data, WWW2012
Q_5	Selver Softic, Semantic Web, Michael Hausenblas
Q_6	Selver Softic, linked data, Information Retrieval
Q_7	Laurens De Vocht, Selver Softic
Q_8	Laurens De Vocht, Selver Softic, 2011
Q_9	Laurens De Vocht, linked data, WWW2013
Q_{10}	Chris Bizer, WWW2013, ISWC2010

Table 5.1: Selected queries by test-users, keywords matching to loaded user profiles are underlined.

of queries Q.

$$AvP(q_i) = \frac{\sum_{k=1}^{A_i} P(k) \cdot rel(k)}{\#\ \text{relevant objects}} \tag{5.11}$$

where A_i is the number of actions taken by the user when resolving the query q_i and $P(k)$ is the precision in the result set after user action a_k in search iteration $k-1$ via the interface V and $rel(k)$ equals to 1 if there are relevant documents after a_k and 0 otherwise. As a result, the items contained in $P(k)$ are k (where $k > 0$) steps away from the matched keyword search context items $P(0)$.

$$MAP = \frac{\sum_{q_i \in Q} AvP(q_i)}{\mid Q \mid} \tag{5.12}$$

Queries

For the evaluation, we restricted our tests to 10 queries which are answerable by the data sets we indexed. These are shown in Table 5.1. These queries representatively cover some of the commonly used search terms within a researcher context: Search for an event ($Q_{1,2,3,4,9,10}$), a person, author or group of authors ($Q_{1,5,6,7,8,9,10}$) or scientific resources ($Q_{1,2,3,6,9,10}$).

Each search runs through the scenario: users enter the first keyword and select the matching result that is resolving their search focus at least one step forward. The users view selected results and can expand them at any time except when the research selects the suggestions from a typeahead interface. Parallel with this selecting and narrowing down the scope, our engine finds relations between the resources and reflects the context. Additionally neighbours which match the selection are found. In the case that users logged in via their Twitter account or Mendeley account or both at same time, their profiles of researcher personalize the boundaries of the search space.

Experimental Setup

We will now explain the experimental setup we have deployed for our system implementing the presented search infrastructure.

Table 5.2 highlights statistics on the used datasets.

Dataset	#Triples	#Instances	#Literals
DBpedia	332 089 989	27 127 750	161 710 008
DBLP (L3S)	95 263 081	13 173 372	17 564 126
COLINDA	143 535	15 788	70 334
Social LD	41 438	7 344	15 350

Table 5.2: Datasets used in the search experiments.

Index Configuration. Table 5.3 shows the statistics of the size of the indexed data sets. The total time for building all indices for all the data sources is about 6 hours. Throughout all the experiments, we use a 8-core single-machine server with 16GB RAM running Ubuntu 12.04 LTS.

Index	#Resources (K)	Temp Space (MB)	Size (MB)
DBpedia	28 384	38 000	30 000
COLINDA + DBLP (L3S)	3 307	15 000	12 000
Social LD	7	5	170

Table 5.3: Resulting index properties based on input datasets.

To ensure maximal scalability and optimally use available resources, we primarily use simple, but effective measures based on topical and structural features of the entities in the search engine. Relations are only computed between pairs in a subgraph of the larger dataset. Every resulting relation as a path between entities are examined for ranking. Only entities belonging to a specific search context are requested. Since the result set of entities might be very large, this “targeted” exploration of relations is essential for the efficiency and scalability.

Resource Alignment. Our earlier results of a case based study, containing several types of user profiles using Twitter and Mendeley to varying degrees, indicate sensitivity, precision and accuracy when linking tags, authors and articles to conferences [6]. Conference tags were better recognized than other tags, this is not surprising because we optimized our model for this task. We never obtained false positives when interlinking authors and articles. When we interlinked followed users on Twitter as authors, we encountered a high amount of negatives. All found links of users as authors were correct but there is room for reducing false negatives.

Efficiency

In order to measure the efficiency of our approach, we stored data about all executed queries: source, destination, all the hops of the path with the links between them and the execution time. We qualify the combined datasets and our algorithm by measuring the average path length and the resolved paths. A found path is relevant if it belongs or has entities relevant to the search context. We measured the hit-rate, distribution of execution time and path lengths for a test set. We compared the results with some metrics used when developing the pathfinding algorithm [5].

Quality. The queries Q_1 to Q_{10} were translated into 576 pathfinding queries between pairs of resources and 400 of those were connected. About 76% were found with the time frame of the evaluation (5 minutes), which is high, considering the relatively small number of resources that had to actually be checked compared to the size of the entire dataset (31.6M resources). Checking a resource means retrieving the resource from the index and identifying the linked resources (neighbours).

The length of the calculated paths is between 0 and 8 hops, a clear majority is between 4 and 6 hops as shown in Figure 5.7. Paths of length 3 and 5 are exceptionally low represented. This is due to the focused nature of the search queries and the resulting manageable number of pathfinding queries. It seemed that the majority paths always go via some publication (publication - author - other publication), which besides a direct link between the resources almost always leads to an even number of hops. This is something particular for the dataset structure and the fact that people mostly look how authors are related. Therefore the majority of paths will have this pattern as a structure. It would not really make sense to consider the average path length, however it is very close to 4.

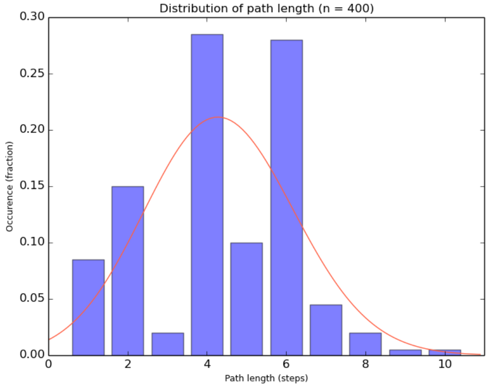


Figure 5.7: There are an unexpected low number of paths with length 3 and 5.

Complexity. Figure 5.8 and Figure 5.9 show respectively the time and space complexity. Except for paths with a length of 3, the average complexities do not increase obviously linearly or exponentially. The engine performs differently than in Chapter 6 because of the multiple datasets we loaded compared to testing with only a single index (DBpedia) loaded [5]. The current results were more volatile and had the pinpointed unexpected deviations with path lengths 3 and 5 from what was expected. This is likely because: (i) the queries were not randomly chosen, (ii) the number of queries was much smaller, and (iii) the dataset is not homogeneous. Some

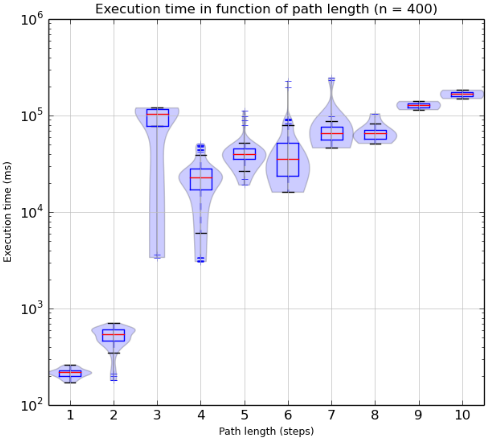


Figure 5.8: Time complexity on a logarithmic scale

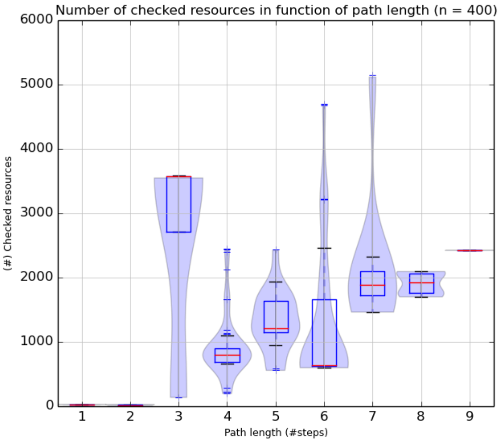


Figure 5.9: Space complexity

paths hop between datasets while others do not. These peculiarities could not occur in the original evaluation. This finding is neither 'good or bad', but it is relevant to notice that the selection and nature of datasets does impact the distribution of path lengths and influences time and space complexity.

Performance. The performance of the algorithm is promising. Even though the configuration was not optimized for speed, but for quality, and was run for the evaluation on a single server only, the algorithm found over 25% of paths in a couple of seconds. Within 30 seconds it found already results for over 50% of the path queries. This is fair as a tolerable waiting time for users is about two seconds [14], but there is room for improvement as the more complex queries take more time to execute. Resolving a keyword and retrieve the matching entities happened instantly. Figure 5.10 shows distribution of the execution times. The search interface and the search engine execute the necessary queries asynchronously and in parallel. While executing the queries – and early results are coming in – the user can immediately start exploring.

Effectiveness

Based upon insights after the first run we reevaluated our system with specific focus on independent judgment of query result and on a comparison to a valid state-of-

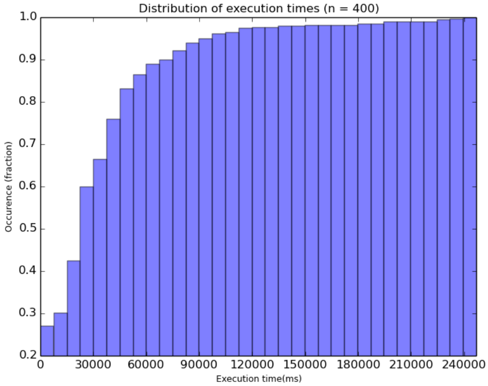


Figure 5.10: More than half of the relations are found in 30 seconds.

the art technology baseline aiming at confirmation of our achieved good results on retrieval.

Baseline. Virtuoso is one of the most common triple stores. It has support for the - non-standard SPARQL - transitive paths and has its own built-in index for text search (via the *bif:contains* property). In many projects dealing with the same amount of data(sets) as we did, it would be the de-facto choice. Therefore we consider it as a baseline for our solution. For the benchmarks we used version *6.1.3127*. We compared executing the 'underlying' queries and the keyword queries in the same way. For example to find non-direct relationships between two resources we used the query as in example Listing 5.1

Two expert-users evaluated independently the results of the baseline. We computed a *F-Measure* (or *positive specific agreement*) of **0.68** and a *chance corrected agreement* (or *inter-rater agreement*) of $\kappa = 0.62$ (where always $-1 < \kappa < 1$). According to Landis et al. [11] this level of agreement is *substantial* to verify that the judgment across both of them is similar enough to be considered.

The mean average precision, **MAP** for the baseline is **0.52**.

The results delivered by baseline approach shown in Figure 5.11 confirm our assumption about very solid retrieval responsiveness with traditional SPARQL queries, however the results from P(2) on are quite low.

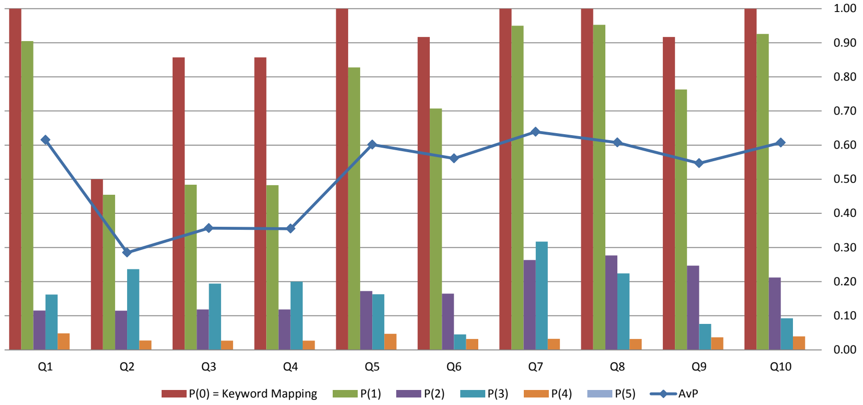


Figure 5.11: Precision results of the baseline for the test queries

```
SELECT ?link ?p ?step ?path
WHERE {
 {
 SELECT ?s ?p ?o
 WHERE {
 { ?s ?p ?o }
 UNION
 { ?o ?p ?s }
 }
 }
 OPTION ( TRANSITIVE,
 t_in(?s),
 t_out(?o),
 t_no_cycles,
 t_distinct,
 t_shortest_only,
 t_min (0),
 t_max (20),
 t_step (?s) AS ?link,
 t_step ('path_id') AS ?path,
 t_step ('step_no') AS ?step,
 t_direction 3
 ) .

 FILTER ( ?o = :Laurens_De_Vocht && ?s = :Selver_Softic )
}
```

Listing 5.1: SPARQL TRANSITIVE Query between resources Selver Softic and Laurens De Vocht

Proposed Engine. To assess the effectiveness of query translation, the same expert users measured the precision and the mean average precision over all queries to evaluate that the search algorithm used in our search engine returns enough high quality relevant results for researchers to achieve their research goals effectively. There was a *F-Measure* (or *positive specific agreement*) of **0.90** and a *chance corrected agreement* (or *inter-rater agreement*): $\kappa = 0.82$. According to Landis et al. [11] this level of agreement is *almost perfect*.

In order to assess our search system we measured the precision of the results for the queries in Table 5.1. To determine the relevance of each resource we relied on expert judgment and we verified expected results against what comes out of the system according to the ranking mechanism. We defined what the expected outcome scenario was based on familiarizing with each of the visualized keyword searches and than having an expert compare the output of the system against the predefined scenario by checking each visualized item one by one after each expansion.

Additionally, we used personalized data to generate a user profile and project the expected search results, see Chapter **??** for more details. This extension is specifically important in the case of the queries with *Selver Softic* and *Laurens De Vocht*, where we loaded these test user profiles. We measured effectiveness using the search interface specified in subsection 5.4 and as described in subsection 5.4.

We judged very precisely each result to enable a more accurate evaluation of the context driven aspect of our search approach. The personalized queries Q_5-Q_9 have been evaluated especially strict. This means that each found resource without direct link to the person, event or topic specified by keyword, are considered a non-relevant result. Even if the resource is relevant in the wider context, for instance a co-author that corresponds to the person but does not fit to the specified event.

Figure 5.12 shows the precision over queries. With exception of Q_1, Q_4, and Q_{10}, queries with preloaded profile data (Q_5-Q_9) deliver more precise results than anonymous queries. This difference is because the main focus of queries Q_5-Q_9 is a person which resolves initially very good within key mapping step, thus following results keep the average precision high. Queries Q_1, Q_4, and Q_{10} have very high precision since they have broader focus which includes more relevant results. The mean average precision **MAP** overall reaches the score of **0.60** which is high but not surprising since the resources within the linked datasets are well-connected and interlinked. The **MAP** we measured in is 8% higher than the baseline case. This first impression strengthens our first evaluation and brings us more near to the confirmation of hypothesis. However to explain the deviation between the results, an additional detailed comparative analysis has to be done. Figure 5.12 shows the precision per query distinguished by path length. As expected the precision decreases with the length of paths. As the path finding progresses over extended links relation to the core concept is becoming weaker. Encouraging however is that the first step of keyword search as well the path finding results of length one, always deliver the results that exceed the value of mean average precision.

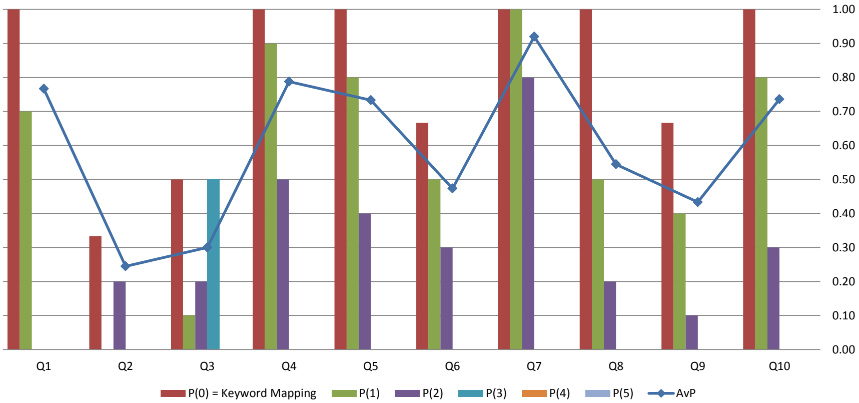


Figure 5.12: Precision of the proposed engine for the test queries over different path lengths

Comparative Analysis. We compared the precision of both sets of results. We have the baseline Virtuoso, which is an integrated system, vs. our proposed semantic engine. While we could just average the expert results or choose on of the results as a reference, we can definitely detect the overall tendencies that reoccur since the inter-rater agreement is sufficiently high, but we can also learn about the cases where they disagree [8]. Therefore, we looked at two scenarios: a strict scenario (both need to agree on relevancy) and a tolerant scenario (at least one needs to judge a result relevant). The results are shown in Figure 5.13 and Figure 5.14.

To be able to compare the results, we included precision until a certain level. Our engine did not contain any items beyond a certain level, $P(3)$. In most cases, this means that the displayed results are all contained within a range of 3 steps from the matched search keyword context. The baseline results are very low from $P(4)$: only a couple of resources at this distance from the search context were considered relevant. At $P(5)$ there are no more results. We choose a strict and tolerant scenario where we either require both experts to judge an element relevant or not respectively. We computed the difference between the precision expressed as $\Delta P = P_{proposed} - P_{baseline}$.

Overall we see the tendency that the proposed engine performs more precise or on par (very little difference) with the baseline, except for Q_5 and Q_8. Q_8 scored bad

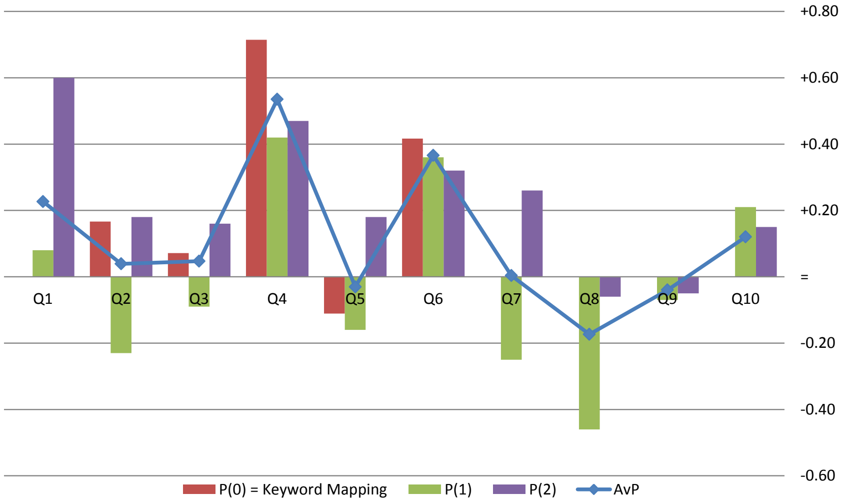


Figure 5.13: The strict delta precision is overall better for the proposed engine except fo Q_5 and Q_8. The better results at $P(2)$ remarkable.

because of the failed interpretation by our engine of *2011*. In either case the precision is there only moderate.

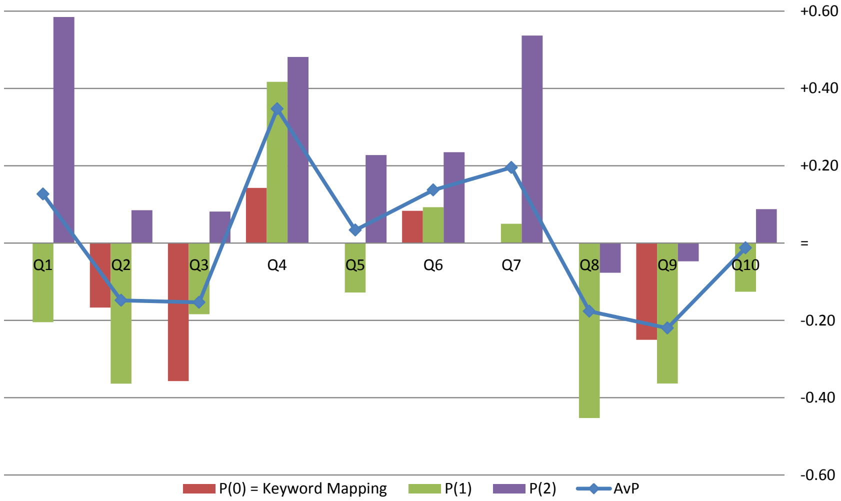


Figure 5.14: The tolerant delta precision is overall similar to the strict delta precision. The proposed engine's precision is less distinct and the results Q_2, Q_3 and Q_9 have less precision compared to the baseline and the strict delta precision.

In the tolerant scenario, we detect overall similar results, but they are more clear-cut, except for Q_2, Q_3 and Q_9. Mainly due to improved results for $P(0)$ and $P(1)$ for the baseline. Q_2, which was on par in the strict case, scores here obviously better for the baseline, particularly because at $P(1)$ one of the reviewers thinks the baseline is more precise. Q_2 is a tricky query because, *ISWC2012* did not take place in *Lyon, France*. The difference is even more distinct for Q_9, as the baseline scored clearly better in the strict case. This is because we found there a larger part of the results at $P(1)$ relevant. Q_9 contains a topic keyword, so it is not trivial for an (expert) user to judge if the results matching this keyword were relevant to both of the keywords. We also see in $Q_2 - Q_3$ that the judgment of $P(0)$ is on par in the tolerant case but much worse for the proposed engine. This is because the expert users did not agree about the relevance of the keyword mapping in the proposed engine. There is an remarkably strong similarity of the results of Q_1, Q_4, Q_6, Q_7, and Q_8. These are also the cases where the proposed engine has the highest precision. This finding is backed with a strong agreement between the raters for both systems.

Summary

In exploratory search scenarios, intermediary link dynamics could lead to relevant discoveries and are therefore not to be neglected. The technique presented in this chapter contributes to data authenticity by guaranteeing that the final output towards the user has useful results in the application domain. Because the technique works with a linked data structure, it is applicable to other domains, if it is structured by adapting the chosen vocabularies according to the datasets used. The evaluation focused not only on pure information retrieval metrics, such as precision (which is more biased towards the final result), but also highlighted how the search effectiveness was gradually influenced by the user's actions.

In terms of *effectiveness*, the proposed engine is more precise than a raw SPARQL baseline for query well-defined contexts, i.e. consist of keywords in which the meaning is unambiguous, for example when a specific conference, author or publication are combined in a search. On the other hand when there are inconsistencies or vague terms, such as topics or years, even mismatches in the query context, expert users disagree about the effectiveness: they judge the relevancy of entities in the results differently.

In terms of *efficiency*, facilitating exploration and search across semantically aligned data sources is feasible as the evaluation showed a linear execution time complexity (scaling with increasing number of hops between resources) and an optimized space complexity. The typical alternative – constructing separate search queries for each of those sources – is a laborious task.

References

[1] B. Aleman-Meza, C. Halaschek-Weiner, I. B. Arpinar, C. Ramakrishnan, and A. P. Sheth. Ranking complex relationships on the semantic web. *Internet Computing, IEEE*, 9(3):37–44, 2005.

[2] K. Anyanwu, A. Maduko, and A. Sheth. Semrank: ranking complex relationship search results on the semantic web. *Proceedings of the 14th international conference on World Wide Web*. WWW '05, pages 117–127, ACM, Chiba, Japan, 2005.

[3] L. Cabral and I. Toma. Evaluating semantic web service tools using the seals platform. *International Workshop on Evaluation of Semantic Technologies (IWEST 2010) at ISWC 2010*. Published in Proceedings of the International Workshop on Evaluation of Semantic Technologies (IWEST 2010): 9th International Semantic Web Conference (ISWC2010), Shanghai, China, November 8, 2010. Edited by Asunción Gómez-Pérez Fabio Ciravegna, Frank van Harmelen, Jeff Hefflin (CEUR workshop Proceedings, Vol. 666, ISSN 1613-0073), 2010.

[4] M. Daoud, L. Tamine, and M. Boughanem. A personalized graph-based document ranking model using a semantic user profile. In: *User Modeling, Adaptation, and Personalization*, pages 171–182. Springer, 2010.

[5] L. De Vocht, S. Coppens, R. Verborgh, M. Vander Sande, E. Mannens, and R. Van de Walle. Discovering meaningful connections between resources in the web of data. *Proceedings of the 6th Workshop on Linked Data on the Web (LDOW2013)*, Rio de Janeiro, Brazil, 2013.

[6] L. De Vocht, S. Softic, E. Mannens, M. Ebner, and R. Van de Walle. Aligning web collaboration tools with research data for scholars. *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*. WWW Companion 2014, pages 1203–1208, International World Wide Web Conferences Steering Committee, Seoul, Korea, 2014.

[7] R. Delbru, S. Campinas, and G. Tummarello. Searching web data: an entity retrieval and high-performance indexing model. *Web Semantics: Science, Services and Agents on the World Wide Web*, 10:33–58, 2012.

[8] T. Demeester, R. Aly, D. Hiemstra, D. Nguyen, D. Trieschnigg, and C. Develder. Exploiting user disagreement for web search evaluation: an experimental approach. *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, pages 33–42, 2014.

[9] K. Elbedweihy, S. N. Wrigley, F. Ciravegna, D. Reinhard, and A. Bernstein. Evaluating semantic search systems to identify future directions of research. R. Garcia-Castro, N. Lyndon, and S. N. Wrigley, editors, *Second International Workshop on Evaluation of Semantic Technologies*. CEUR Workshop Proceedings 843, pages 25–36, May 2012.

[10] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design Patterns: Elements of Reusable Object-oriented Software. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

[11] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 1977.

[12] G. Marchionini. Exploratory search: from finding to understanding. *Commun. ACM*, 49(4):41–46, April 2006.

[13] J. L. Moore, F. Steinke, and V. Tresp. A novel metric for information retrieval in semantic networks. *The Semantic Web: ESWC 2011 Workshops*. Springer, pages 65–79, 2012.

[14] F. F.-H. Nah. A study on tolerable waiting time: how long are web users willing to wait? *Behaviour & Information Technology*, 23(3):153–163, 2004.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: bringing order to the web. Technical Report 1999-66. Previous number = SIDL-WP-1999-0120. Stanford InfoLab, November 1999.

[16] X. Pintado. Object-oriented software composition. In. Prentice Hall PTR, 1995. Chap. The affinity browser, pages 245–272.

[17] D. M. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[18] M. Schmidt, T. Hornung, G. Lausen, and C. Pinkel. Sp2Bench: a sparql performance benchmark. *IEEE 25th International Conference on Data Engineering*. Volume abs/0806.4627 of 2008.

[19] V. Uren, Y. Lei, V. Lopez, H. Liu, E. Motta, and M. Giordanino. The usability of semantic search tools: a review. *The Knowledge Engineering Review.* 22(4):361–377, December 2007.

Chapter 6

Path-based Storytelling

The computer can't tell you the emotional story.
It can give you the exact mathematical design,
but what's missing is the eyebrows.

—Frank Zappa.

This chapter investigates algorithms to generate semantically annotated paths that 'tell stories', an important technique to explore and discover 'serendipitous', meaningful and non-trivial connections between multiple resources in linked data on the Web. The weights of links in paths and heuristics to order candidate resources form the essential building blocks of the algorithm's architecture and address different aspects of serendipity. Furthermore, optimizations of the base algorithm tweak the link estimation and the selection of resources in a path. Experimental findings with path-based stories in DBpedia indicate the performance of the base algorithm and measure the improvements when applying optimizations.

6.1 Introduction

Path-based storytelling can be seen as a particular kind of querying data. Given a set of keywords or entities, which are typically, but not necessarily dissimilar, it aims at generating a path by explicitly relating the query context with a path that includes semantically related resources. The semantic relations in linked data between a single chain of links (or nodes) define how two concepts are related to each other. For example, in DBpedia[1], we can find associations being a direct link such as **Paris** is the *capital of* **France**; but also longer chains such as **Paris** is the *birthPlace* of **Martine Aubry**, which is the *successor* (as First Secretary of the Socialist Party) of **Francois Hollande**, which is the *president* of **France**. Any single chain of links preserves the value of well-assigned information fitting to a context and concepts of the underlying graph. This kind of search and exploration algorithms serves the objective of qualitative informational retrieval and knowledge discovery.

Relating chains of indirectly connected resources through paths provides users the ability to explore concepts in a non-traditional, more entertaining and educational way. Many state-of-the-art pathfinding approaches aim at combining sequences of resources that coincide with the user's expectations. According to Heim et al. "real discovery is only possible with a human involved, since only the user can ultimately decide if a found relationship is relevant in a certain situation" [27]. Graph algorithms are designed to make optimal use of available computation resources to find paths in structured data in a variety of applications (e.g. navigation systems). Applying them to linked data can facilitate the resolution of complex queries that involve the semantics of the relations between resources. In doing so, it is a challenge to improve and tailor existing approaches to match user expectations so that users are able to explore relevant data as much as possible. The evaluated measures (performance, semantic relatedness, and relevance) for the base implementation and optimization of the proposed algorithm address research questions **RQ1** and **RQ3**.

Motivating Examples

Finding precise relations within chains of linked entities is not only interesting for semantic search in fact-based knowledge repositories or digital archives. Path-based storytelling is also applied for example in entertaining applications and visualizations [40] to enrich related linked data resources with data from multimedia archives and social media [14], as well as in scientific research fields such as bio-informatics where biologists try to relate sets of genes arising from different experiments by investigating the implicated pathways [30], discovering stories through linked books [10], or refining event contexts in named entity recognition [39].

[1] `http://dbpedia.org`

Serendipity

The aspects that make a path or story ‘relevant’ are captured in the term *serendipity*. The term depicts “a mixture between casual, lucky, helpful, and unforeseen facts, in general but also in an information context” [21]. This means that a path-based story should be relevant but include things that the user did not expect. In fact, when users during exploration not only want to confirm, but also extend knowledge, discover and even be surprised, they do not want to feel unsure while doing so. This means that users can always relate presented facts to their background knowledge.

Semantics

When the links between nodes are semantically annotated, such as in large real-world linked data graphs, users are able to directly interpret the transitions between nodes and thus the meaning of a path. Applying pathfinding approaches increases arbitrariness due to the large number of possible relations that connect two entities in a query context. This arbitrariness becomes clearly visible precisely because of the semantics. Even optimal paths frequently show a high extent of arbitrariness caused by the inevitable increasing number of nodes and sometimes loosely related links among them: paths appear to be determined by chance and not by reason or principle and are often affected by resources that share many links. In addition, large real-world linked data graphs typically exhibit small-world properties. This means that the graphs often consist of sub-graphs which have connections between almost any two nodes within them and contain nodes with a very high degree. For example, countries are nodes with a very high degree: every person and city link to the country they belong to and countries are frequently linked to other countries which then link further to other persons and cities.

Applying Graph-based Algorithms

Applying path algorithms to linked data has the advantage that the links between nodes are annotated, thus introducing semantics. This allows interpretation of the transitions between nodes and the meaning of each path. It is not trivial to rely on linked data queries when designing an algorithm to find path-based stories in linked data graphs. State-of-the art RDF stores like Virtuoso or Allegrograph [33] or graph databases like Neo4J are not designed specifically for this purpose, but it may be argued that they provide API’s that would allow for the development of algorithms that work on top of them. In that case, the database delivers only the functionality to do local exploration, but the intelligence has to come from the path algorithm. SPARQL is not able to query for arbitrary paths, it is currently only possible to check for the existence of an arbitrary connection, so-called 'property paths' (SPARQL 1.1).

Most implementations of pathfinding algorithms are application specific, for instance routing in navigation systems for vehicles [4, 32]. Pathfinding is a well known issue in graph theory and mathematics [7]. It refers to finding a path between two nodes in a graph. Various algorithms have been described to solve this issue in graphs. The two most common algorithms are Dijkstra [18] and A* [24]. The former finds a path by selecting nodes with the shortest distance to the source. This distance is calculated using the weight of the edges, resulting in the optimal path. The latter extends Dijktra's algorithm with a minimal approximated remaining distance, based on a provided heuristic, between a node and the end node. This allows the algorithm to evaluate less nodes, which increases its performance. During the execution of the algorithm evaluating a node is an expensive operation because data needs to be retrieved and checked. If the data needs to be retrieved via the web, this causes an additional delay making the operation even more expensive. A* commits to be memory efficient, but that does not mean at larger distances it is still very complex to find paths. Nevertheless, adequately limiting of the search scope a priori and fitting weights and heuristics at runtime, contribute to the overall execution.

Cui and Chi [9] reviewed A*-based algorithms and techniques from different perspectives but did not investigate the semantics. Eliassi-rad and Chow [20] used ontological information, probability theory, and heuristic search algorithms to reduce and prioritize the search space between a source vertex and a destination vertex. They developed two heuristics for semantic graphs to be used with the A* algorithm. In the biomedicine domain, He et al. [26] demonstrated how graph-theoretic algorithms for mining relational paths can be used together with Chem2Bio2RDF [6] data to extract new biological insights about the relationships between its data. They presented a scalable path finding algorithm that works on RDF to find complex relationships between biological entities, e.g., genes, compounds, pathways, and diseases. Pathfinding has been performed in metabolic graph by searching for one or more paths with lowest weight. The weights assigned to each compound were the number of reactions in which it participates.

A* is based on a graph representation of the underlying data (i.e., resources and links between them define nodes and edges, respectively) and determines an optimal solution in form of a lowest-cost traversable path between two resources. The optimality of a path, which is guaranteed by the A* algorithm, does not necessarily comply with the users' expectations [13].

Presenting Paths as a Story

Each path will contain multiple facts that may contribute to a story. This is because each step in the path is separated with at least one hop from the next node. For example, to present a story about *Carl Linnaeus* and *Charles Darwin*, the story could start from a path that goes via *J.W. von Goethe*. The resulting statements serve as basic facts, which are subject-relation-object statements, that make up the story.

A set of statements is not a presentable story. The story's statements may originate from multiple paths. It is up to the application or visualization engine to present it to end-users and enrich it with descriptions, media or further facts. Table 6.1 exemplary explicates the idea of statements as story facts.

Table 6.1: The statements from which a story can be generated.

About	**Relation**	**Object**
Carl Linnaeus and Charles Darwin	are	scientists
J.W. von Goethe	influenced	Carl Linnaeus and Charles Darwin
J.W. von Goethe and Charles Darwin	influenced	Karl Marx and Sigmund Freud

6.2 Architectural Model

A component for path-based storytelling, shown in figure 6.1, consists of a 'pathfinder' that focuses on the execution of the algorithms. The pre-processor takes care of selecting the required data and transforming it to the adequate data structure. The post-processor handles the results of the pathfinding algorithm and prepares them for further handling by the bridge component taking care of the query processing.

The proposed base algorithm, which is given in Algorithm 1 takes a start and destination resource as inputs, and returns a possible path between them. It consist of two parts: pre-processing and graph browsing. This approach enables finding paths in linked data graphs and makes use of the A* algorithm during the iterate step.

The algorithm uses the blackboard design pattern, which provides a computational framework for the design and implementation of systems that need to integrate

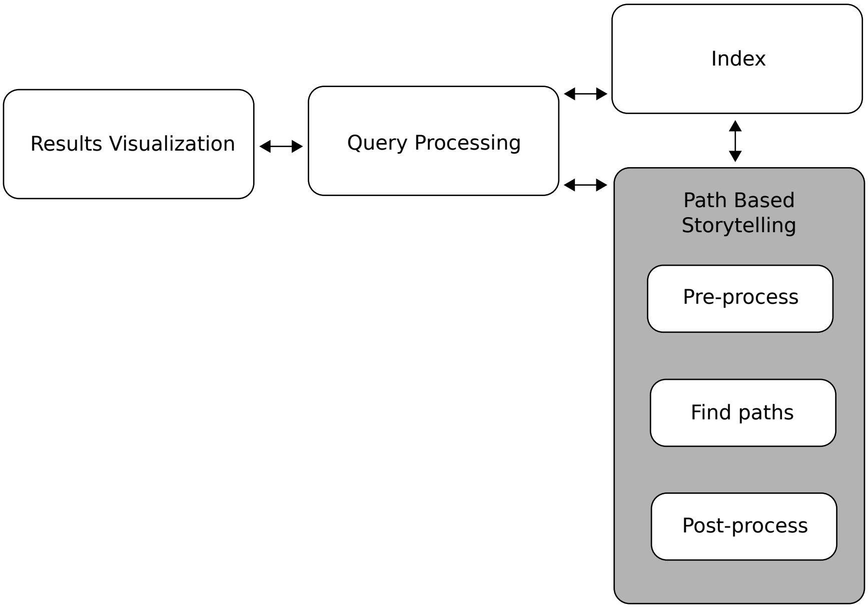


Figure 6.1: Three main modules for path-based storytelling: pre-processor, pathfinder and post-processor.

large and diverse specialized modules, and implement complex, non deterministic control strategies [31]. Additional steps make the approach work because the A* algorithm is memory intensive:

- Pre-processing is required to generated the indexed linked data.
- After the initialization, the part of the graph being considered for inclusion in the blackboard, grows with each iteration and it is decided if the search should continue. In the blackboard the graph is represented as an adjacency matrix. Section 6.2 goes into details about this process.

Figure 6.2 shows how the blackboard is used for the execution of the A* algorithm. The figure shows the points where the base algorithm has to decide upon the inclusion or exclusion of certain resources to make the algorithm work and deal with memory issues. Each iteration follows these steps:

1. filtering the input graph;
2. rank reduction of the graph's adjacency matrix;

Data:
start: *source*
destination: *target*
Result:
path between source and target

```
adjacency_matrix = initialize(start,destination)
iteration = 0
path = False
stop_condition = not path and iteration < MAX
while stop_condition:
 path = iterate(adjacency_matrix)
 iteration += 1
termination(path)
```

Algorithm 1: The algorithm iterates over the adjacency matrix until the stop condition is met.

3. check if a path exists;

4. if no path exists: (a) reiterate, or (b) compute the path ;

5. (a) if no path is found the graph will be expanded; (b) if a path is found, the algorithm will determine the rank of the path: if the ranking score of the highest ranking path is sufficient terminate, otherwise reiterate.

Pre-processing

The algorithm converts the source data to lists of triples and groups them in documents per subject and loads those documents into an index. Figure 6.3 shows this for example for a graph containing resources related to *France* and *Germany*. The index contains references (URIs) to all the considered candidate-resouces. The index is an efficient method to instantly retrieve a resource given a match pattern.

Finding Paths

Given the index, source and destination, the base algorithm outputs the path between source and destination nodes as a list of all the URIs and the predicates connecting them. The algorithm iterates over a growing pool of candidate resources that might lead to a path. Figure 6.4 shows that during the process of finding paths

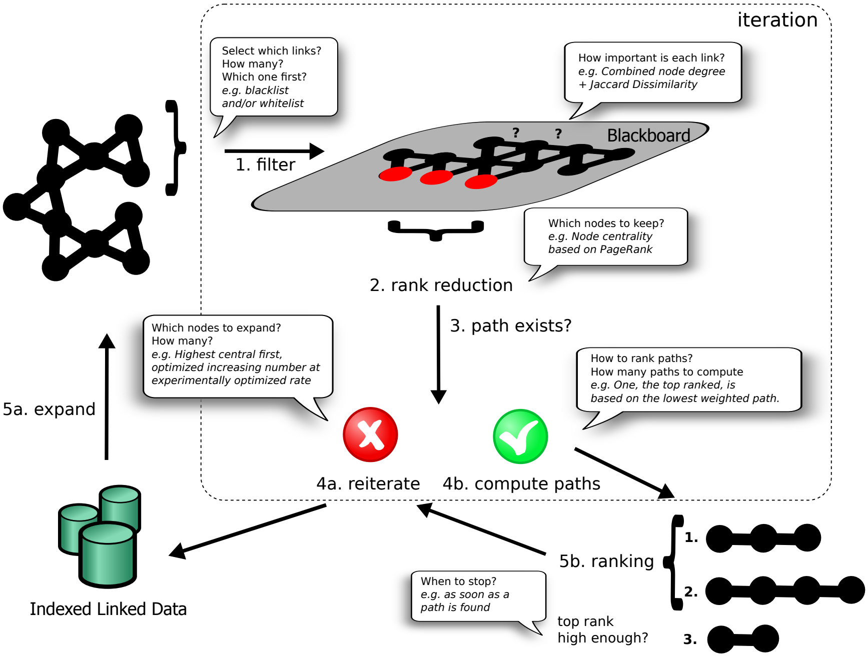


Figure 6.2: The overview of an iteration of the graph-based storytelling approach, after initialization. The questions that the algorithm 'asks' at each step during the process are shown in the text bubbles.

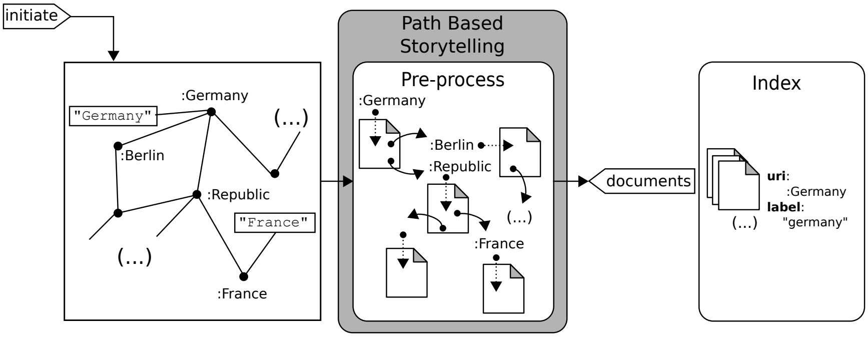


Figure 6.3: Pre-processing resources

between *France* and *Germany*, the pool of nodes taken into consideration is expanded on both sides. The links between candidate resources are verified against a list of acceptable paths. This ensures quality of the paths and avoids senseless or trivial connections between the resources. The users are only interested in meaningful links, so the algorithm makes optimal use of the semantic properties of each re-

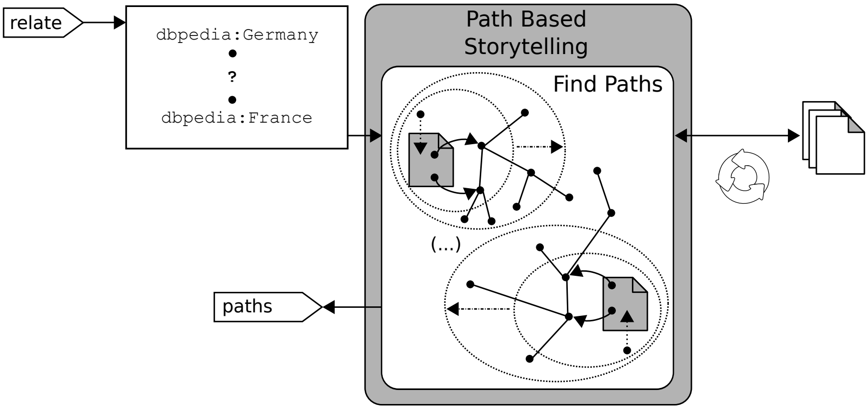


Figure 6.4: Finding paths between resources. The dashed ellipses show the expanding pool of candidate nodes starting from the indexed documents (marked grey) of the start and the destination node.

source. The base algorithm consists of three main steps: initialization, iterations, and termination.

Initialization. The algorithm fetches all the children of the start node, named *source*, and the destination node, named *target*. A global set containing references to all resources is retained, as in for example Table 6.2. In the example is *Paris* the source and *Barack_Obama* the target. Next, all the children of *source* and *target* are stored in the global set with references to the original resources. Table 6.3 shows that each child is also a set with the resources as keys, and that the predicates are linked to it as values.

Resources
:Paris
:Barack_Obama
:France
:Eiffel_Tower
:United_States

Table 6.2: Example global set of resources

These data structures are converted to an adjacency matrix. The adjacency matrix represents which resources have a direct link to each other. The use of an adjacency matrix allows implementing A* with low-level math libraries. A list with positions

:Paris	
:France	:capital
:Eiffel_Tower	:monument

Table 6.3: Example resource with predicates and objects

corresponding to the row and column numbers refers to the resources stored inside the global set. The resources are kept also in a list with index positions:

$resources = (0 = Paris, 1 = Barack\_Obama, 2 = France, 3 = Eiffel\_Tower,$
$4 = United\_States)$

The positions in the list *resources* correspond with the rows and columns of the adjacency matrix in Table 6.4.

The adjacency matrix is a symmetrical sparse matrix, as most of the cells are 0. Most of the cells are 0 because there is no direct link between most resources. Note that index does not distinguish between forward and backward links. This has the benefit of resulting in a symmetrical matrix. For example: *France* is linked to *Paris* as "has capital" and the inverse link "is capital of" is equally important. Only when there is a parent-child connection or vice-versa a cell gets value 1. Links between the same resources are ignored, resulting in a value of 0 in the matrix, to avoid loops.

$$\left[\begin{array}{c|ccccc} * & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 0 & 0 \\ 4 & 0 & 1 & 0 & 0 & 0 \end{array}\right]$$

Table 6.4: Row and column 0 show a link with row and column 2 and 3 which correspond in the list *resources* with *Paris*, *France* and *Eiffel_Tower* respectively.

Iterations. Each iteration executes the A* algorithm on the resources that made it in the blackboard during this iteration. Before the actual execution, each link between nodes is assigned a weight using a Dice-based [17] semantic weight measure:

$degree(node) = sum(nodelinks)$

$weight(parent, child) = log(degree(parent) + degree(child))$

This semantic weight measure is perfectly suited as a metric for weighting the paths. It was introduced to optimize the quality of the links. Rare nodes, nodes with low probability that a random walk returns to the same node, lead to better and more interesting paths. It was shown that a weight as the sum of the links of each node is a valid measure for this [36].

A* requires a heuristic for estimating the distance between nodes. This allows the sorting of the links in order of probability of leading to a path, without having to calculate the actual distance, resulting into a performance gain. As a suitable heuristic, we selected the Jaccard distance. The Jaccard distance measures dissimilarity between sample sets and is complementary to the Jaccard similarity coefficient. The Jaccard similarity coefficient is one of the most efficient measures for semantic relatedness [29]. If nodes share a lot of the same predicates, we assume that they are closely related to each other. This makes it very likely to find a path between them. We obtain the Jaccard distance by dividing the difference of the sizes of the union and the intersection of two sets by the size of the union. The sets contain the predicates of each node ($node_x$ = set of predicates in node x).

$$jaccard_{distance}(node_A, node_B) = \frac{\|node_A \cup node_B\| - \|node_A \cap node_B\|}{\|node_A \cup node_B\|}$$

Once we have defined the weights for each link and defined our heuristic, we try to find a path in the pool of resources using the adjacency matrix provided. If no path is found, we find the children of the bottom level nodes and add them to the set of resources. They will be used in the next iteration. We update the existing parents of all generations to see if there are any links to the newly added nodes. The child resources added in each iteration form a generation. If we have found a path the algorithm terminates.

Termination. A stop condition prevents the algorithm from running indefinitely when no path is found. Since it is unlikely to find a path if no path has been found after a while, the algorithm should stop. Therefore the algorithm can be configured to stop after a limited: amount of iterations, execution time, number of checked resources. This limit depends on the dataset and the target application. Another possibility is to consider a threshold on the found paths: continue until a path that obtains a certain score or rank.

6.3 Implementation

This section explains the implementation of the base algorithm for path-based storytelling corresponding to the architectural model outlined in section 6.2. As mentioned in section 6.1, searching for relationships between Linked Data resources is typically interpreted as a pathfinding problem: looking for chains of intermediary nodes (hops) forming the connection or bridge between these resources in a single dataset or across multiple datasets. Linked Open Data, linked datasets available via the web, introduce challenges for pathfinding algorithms. In many cases centralizing all needed linked data in a certain (specialized) repository or index to be able to run the algorithm is not possible or at least not desired [15]. An optimized version of the base algorithm is introduced to improve the serendipity of paths and make the implementation more web-oriented and reduce the dependency on the custom index. This effectively eliminates the pre-processing step required by the base algorithm by implementing the optimized algorithm on top of triple pattern fragments [41]. This allows to use the implementation in combination with any triple pattern fragments compatible web server.

Base Algorithm

To implement the framework, we first indexed DBpedia and tested the performance of a test set with random queries. The base algorithm was implemented using Python with Numpy and exposed as a REST Service[2] with Linked Open Data[3] extracted from Wikipedia: the DBpedia dataset [5]. DBpedia defines linked data URIs for millions of concepts. Many other initiatives create links from their datasets to DBpedia, making DBpedia the most centralized dataset on the Web.

The index speeds up the information retrieval process and allows processing hundreds of match requests on the graph per second. To be able to achieve this performance, the data structure of each index entry needs to be optimal. SPARQL endpoints and RDF stores are only scalable to a certain degree and the query time depends on the size of the dataset [25, 33]. For the combination of frequency and type of queries needed for our algorithm none of the current SPARQL endpoints was suitable. At the time of implementing the base algorithm, "Semantic Information Retrieval Engine" (SIREn) was a popular semantic index and proved to be the most

[2] Pathfinding Service. `http://pathfinding.restdesc.org` - last accessed: March 2017

[3] Open Data represented as RDF

adequate solution for our algorithm. SIREn started as a specialized SOLr [4] extension for linked data [16] and was later released in 2013 as a search solution for (JSON) semi-structured documents in general [5]. SOLr is a HTTP layer over Lucene, the well-known indexing system for textual data lookup. SIREn extends SOLr to allow indexing and querying of linked data resources.

The time to create an adjacency matrix increases exponentially and the required memory space quickly hits machine limits. We noticed that this is due to the adjacency matrix becoming too large. To avoid that the adjacency matrix becomes too large to process, we ensure a limited number of resources to check while still increasing the probability of finding a valid path with each iteration. To increase the probability of finding a path with each new iteration, we estimate which resources are the most important and drop those who are not. Important nodes have the highest probability of leading to a path and thus have links to many other important nodes. Thus, we link resources to as many as possible related resources that are again linked to a lot of highly linked resources (hubs). We do not distinguish between outgoing and incoming links. All relations in linked data have an inverse that is equally important. Both hubs and resources that receive a lot of incoming links (authorities) behave the same in the algorithm. This is because all links are reversible (as explained earlier in the architectural model section 6.2).

We can find a reduced rank approximation to the adjacency matrix by setting all but the first k largest singular values equal to zero and using only the first k columns of the resulting decomposed matrices. We get the singular values through Singular Value Decomposition (SVD). Though our adjacency matrices were sparse, we noticed that the required SVD performs slowly. SVD requires a complete dataset, and has significant memory requirements. The SVD leads to Hyperlink-Induced Topic Search (HITS) (also known as hubs and authorities), a link analysis algorithm.

Another centrality measure is the PageRank algorithm, it reflects the so-called random surfer model, meaning that the PageRank of a particular page is derived from the theoretical probability of visiting that page when clicking on links at random. However, real users do not randomly surf the web, but follow links according to their interest and intention. A page ranking model that reflects the importance of a particular page as a function of how many actual visits it receives by real users is called the intentional surfer model.

[4] http://lucene.apache.org/solr/

[5] http://rdelbru.github.io/SIREn/

The difference between the two approaches mentioned above is that SVD / HITS uses singular values while PageRank uses eigenvalues [19]. HITS emphasizes mutual reinforcement between authority and hub webpages, while PageRank emphasizes link weight normalization and node hopping based on random walk models. We did not look into hybrid or unified approaches because it was out of scope and PageRank and HITS, lead to similar ranking of the nodes. We were thus convinced that it was fast enough and guaranteed a good ranking of nodes. Initially, the algorithm ordered nodes according to node centrality with SVD at first, but quickly the intensive memory requirements became clear. A sparse matrix iterative numerical optimization of SVD and HITS was much faster but did not converge to a solution frequently enough. PageRank on sparse matrices performs in this case faster, the iterative implementation always converges and produces a ranking of the nodes that guarantees that the most important nodes stay in the candidate pool each iteration. We also tested simply ignoring the nodes below a certain threshold, with less than a fixed number of links. It was much faster to compute, but it did not introduce a more densely linked node pool with each iteration. This is because keeping nodes with many links to nodes with few links is not really interesting and results in a node pool with too many unimportant nodes compared to the node centrality based approaches.

Optimized Algorithm

Each path is determined within a query context comprising both start and destination resources as is the case for the base algorithm. The optimized algorithm reduces the arbitrariness of a path between these resources by increasing the relevance of the links between the nodes using a domain-delineation step. The path is refined by iteratively applying the A* algorithm and with each iteration attempting to improve the overall semantic relatedness between the resources until a fixed number of iterations or a certain similarity threshold is reached.

Domain Delineation. Instead of directly initializing the graph as-is by including all links between the resources, we identify the relevance of predicates with respect to the query context. This is done by extracting and giving higher preference to the type of relations (predicates) that occur frequently in the query context. In this way, we make sure that the links included in the relation matter because each predicate that describes the semantics of a link also occurs in the direct neighborhood

```
Data: start, destination, graph, k
Result: list of important predicates given the context
initialize pf_irf_p_list;
predicates_start = unique predicates start;
predicates_dest = unique predicates destination;
predicates_considered = intersection predicates_start predicates_dest;
foreach predicates_considered as p do
    pf_irf_p = compute pf_irf p;
    add pf_irf_p to list
end
reverse sort pf_irf_p_list;
take the first k elements of the list as important predicated;
```

Algorithm 2: Selecting important predicates

of the query context. The selection of the most important predicates for domain delineation is shown in Algorithm 2.

An adapted variant of the TF/IDF [1] measure, 'PF/IRF' orders the links in a graph to select the ones that are the most relevant based on the given start and destination nodes. The PF/IRF measure reflects the importance of a predicate with respect to a resource in a dataset and is defined as follows:

$$PF(p) = \frac{\text{Number of times predicate } \mathbf{p} \text{ appears in a resource}}{\text{Total number of predicates linked to the resource}} \tag{6.1}$$

$$IRF(p) = \ln \frac{\text{Total number of resources}}{\text{Number of resources with predicate } \mathbf{p} \text{ in it}} \tag{6.2}$$

For example, the PF/IRF computation for predicates linked to *Carl Linnaeus* is explained below for the case when PF/IRF is determined in the context of start *Carl Linnaeus* and destination *Charles Darwin* based on DBpedia.

1. We determine predicates that are important in the context. This is done by retrieving the distinct predicates that are linked to the context nodes.

2. For each predicate, we compute its occurrence based on linked nodes. In addition, the total number of predicates linked to the resource *Carl Linnaeus* is determined.

3. As a result, the total number of predicates linked to the resource *Carl Linnaeus* is 9890. For the predicates *binomialAuthority* and *label* we obtain the values

2297 and 12, respectively. The total number of resources (including objects) in the DBpedia is $M = 27,318,782$.

4. We compute the number of resources which are linked using each predicate by counting the distinct number of resources through the predicate *binomialAuthority* and *label* in both directions. This results in $155,207$ and $10,471,330$ respectively.

5. By using the PF/IRF formula above we finally get the following values:

   $\mathrm{PF/IRF}(binomialAuthority) = 2297/9890 * \ln(27,318,782/155,207) =$ **1.20** and $\mathrm{PF/IRF}(label) = 12/9890 * \ln(27,318,782/10,471,330) =$ **0.0011**

Since the PF/IRF value of *binomialAuthority* is much higher than that of *label*, the predicate *binomialAuthority* is more likely to be included.

Algorithm. The output of the aforementioned domain delineation step can be thought of a linked data graph comprising nodes and predicates which are semantically related to the user's query context. To provide a serendipitous relation based on this linked data graph, the graph has to be traversed via a meaningful path including the start and end destination of the query context. A single or multiple paths are then used as essential building blocks for generating a relation.

To find a path in a linked data graph, we use the A* algorithm due to its ability of computing an optimal solution, i.e., a (shortest) cost-minimal path between two nodes with respect to the weights of the linking predicates contained in the path. To reduce the number of predicates to be examined when computing the lowest-cost path between two nodes and, thus, to achieve an improvement in the computation time of the A* algorithm, heuristics are frequently used to determine the order of expansion of the nodes according to the start and end node provided within the query context. In addition to a heuristic, the A* algorithm utilizes a weighting function to determine paths which are semantically related to source and destination nodes as specified within the query context. Thus, the serendipity of a relation generated based on a single or multiple paths is strongly connected to the underlying weighting scheme and heuristic. In the following section, we propose and investigate various heuristics before we will introduce different weighting schemes.

Refinement. After a path is determined by the A* algorithm, we measure the semantic relatedness, corresponding to the lowest semantic distance between all resources occurring in the path with respect to the query context. This is done for example by counting the number of overlapping predicates (i) among each other combined with those in the start and destination resources; and then (ii) averaging and normalizing this count over all resources. Depending on the threshold and the maximum number of iterations configured, this process is repeated, typically between 3 and 10 times. Finally, the path with the shortest total *distance* (or cost) is selected for the relation. The *distance* for a $path = (s_1, s_2, ..., s_n)$ is computed based on a weight function w as distance$(path) = \frac{\sum_1^{n-1} w(s_i, s_{i+1})}{n}$.

Heuristics

The objective of a heuristic is to determine how a node in a linked data graph is semantically related to the query context, i.e. source and destination nodes, and thus a good choice for expansion within the A* algorithm. For this purpose, we formally define a heuristic as a function $heuristic : G \times G \rightarrow \mathbb{R}$ that assigns all pairs of nodes $n_a, n_b \in G$ from a linked data graph G, a real-valued number indicating their semantic relation [12].

Jaccard Distance. The first heuristic we consider is the **Jaccard distance** which is a simple statistical approach taking into account the relative number of common predicates of two nodes. The higher the number of common predicates, the more likely similar properties of the nodes and thus the semantically closer in terms of distance the corresponding nodes. The Jaccard distance $jaccard : G \times G \rightarrow \mathbb{R}$ is defined for all nodes $n_a, n_b \in G$ as follows:

$$jaccard(n_a, n_b) = 1 - \frac{\|n_a \cap n_b\|}{\|n_a \cup n_b\|} \tag{6.3}$$

Normalized DBpedia Distance. Another approach that can be utilized as a heuristic is the **Normalized DBpedia Distance** [11, 23]. This approach adapts the idea of the Normalized Web Distance to DBpedia and considers two nodes n_a and n_b to be semantically similar if they share a high number of common neighboring nodes linking to both n_a and n_b. The Normalized DBpedia Distance $NDD : G \times G \rightarrow \mathbb{R}$ is defined for all nodes $n_a, n_b \in G$ as

$$NDD(n_a, n_b) = \frac{\max(\log f(n_a), \log f(n_b)) - \log f(n_a, n_b)}{\log N - \min(\log f(n_a), f(n_b))}, \tag{6.4}$$

where $f(n) \in \mathbb{N}$ denotes the number of DBpedia nodes linking to node $n \in G$, $f(n,m) \in \mathbb{N}$ denotes the number of DBpedia nodes linking to both nodes n and $m \in G$, and where the constant N is defined as the total number of nodes in DBpedia, which is about $2.5M$.

Confidence. Another heuristic that has been proposed for semantic path search in Wikipedia is the **Confidence measure** [22]. The Confidence measure is an asymmetrical statistical measure that can be thought of as the probability that node n_a occurs provided that node n_b has already occurred. The Confidence measure $P : G \times G \rightarrow \mathbb{R}$ is defined for all nodes $n_a, n_b \in G$ as:

$$P(n_a | n_b) = \frac{f(n_a, n_b)}{f(n_b)} \tag{6.5}$$

As opposed to the heuristics, which affect the expansion order within the A* algorithm by estimating the potential semantic relatedness of a node, weighting schemes are finally utilized to asses the quality of a path. We propose different weighting schemes in the following section.

Weights

The objective of a weighting function is to determine the exact cost of a path, which is the sum of weights of linking nodes. A weighting is formalized as a function $weight : G \times G \rightarrow \mathbb{R}$ between the corresponding nodes from the linked data graph.

Jaccard Distance. We apply the **Jaccard distance** in exactly the same way to determine the weights so that the core algorithm prefers similarity in adjacent nodes in each path. We use this distance between two directly adjacent nodes rather than unconnected nodes in the graph.

Combined Node Degree. Moore et al. [36] proposed the **combined node degree** which can be used to compute a weight that encourages rarity of items in a path. It ranks more rare resources higher, thereby guaranteeing that paths between resources prefer specific relations. The main idea is to avoid that paths go via generic nodes. It makes use of the node degree, the number of in and outgoing links. The combined node degree $w : G \times G \to \mathbb{R}$ is defined for all nodes $n_a, n_b \in G$ as:

$$w(n_a, n_b) = \log(\deg(n_a)) + \log(\deg(n_b)) \tag{6.6}$$

Jiang and Conrath Distance. Mazuel et al. [34] suggest to take into account the object property ontology relation between two adjacent items in a path. The base distance measure there is the **Jiang and Conrath distance** [28], which we can interpret in terms of RDF by looking at the classes of each of the nodes and determining the most common denominator of those classes in the ontology. Once this type is determined, the number of subjects that exist with this type is divided by the total number of subjects. The higher this number, the more generic the class, thus the more different two nodes.

Discussion

It is important to note that the main complexity of the approach is in line with the centrality of underlying graph-indexing and data-processing algorithms. It turns out that server-side query processing degrades the performance of a server and therefore limits its *scalability*. While many approaches are suitable for a small-to-moderate number of clients, they reveal to be a performance bottleneck when the number of clients is increased.

Instead of running the algorithm entirely on the server, we moved CPU and memory intensive tasks to the client. The server translates user queries into smaller, digestible fragments for the data endpoint. All optimizations and the execution of the algorithm are moved to the client. This has two benefits: (i) the CPU and memory bottleneck at server side are reduced; and (ii) the more complex data fragments to be translated stay on the server even though they do not require much CPU and memory resources, but they would introduce to many client-side requests.

A separate index with linked data documents to store the fragments for fast navigating graphs served a first iteration but turned out to be only limited scalable. It required each time a pre-selection of datasets that would need to be manually or semi automatically scheduled to be ingested or updated. The improved algorithm[6] runs using Triple Pattern Fragments (TPF). TPF provides a computationally inexpensive server-side interface that does not overload the server and guarantees high availability and instant responses. Basic triple patterns (i.e. *?s ?p ?o*) suffice to navigate across linked data graphs (no complex queries needed).

6.4 Evaluation

This section explains the evaluation of the base algorithm in terms of performance and result quality. It verifies to which degree the optimized algorithm is able to reduce the arbitrariness of the paths in comparison to the base algorithm. The base algorithm used jaccard and combined node degree heuristic and weight but different heuristic and weights have an impact on the resulting paths. Using the optimized algorithm this impact is investigated.

[6] The base algorithm can be found at https://github.com/mmlab/eice and the improved algorithm at https://www.npmjs.com/package/everything_is_connected_engine

Base Algorithm

To evaluate the base algorithm approach, we store data about retrieved paths: source, destination, all the hops of the path with the meaning of the links between them and the execution time. We check the average length of found paths and we measure the fraction of paths found within various time frames. A found path is relevant if it occurs within a tolerable time for the users. Depending on the context and the size of the dataset this time may vary. We measure the hitrate, distribution of execution time and path lengths for a testset containing 10000 random path calculations randomly among 200 DBPedia resources (popular cities, countries or brands). The total indexed dataset (based on DBPedia version 3.8) contains $10.8M$ resources. We set the stop condition for the algorithm on a path length of 12.

Meaningfulness. The found paths should not be arbitrary, for example Paris and Barack Obama could have been linked because Barack Obama lives in the White House in Washington DC. Both Paris and Washington DC are cities and this would be a very short and relevant path. This is however not that meaningful for most users. Executing a search for path between Paris and Barack Obama gives output as in Table 6.5.

Path		
:Barack_Obama	:isPresidentOf	:Joe_Biden
:Joe_Biden	:religion	:Catholic_Church
:Catholic_Church	:isReligionOf	:Bertrand_Delanoe
:Bertrand_Delanoe	:isMayorOf	:Paris

Table 6.5: Output for the search for a path between Paris and Barack Obama

We observe that the path goes over Bertrand Delanoe and the shared religion with Joe Biden. This is still a simple result but it already exposes a route that is meaningful. This is achieved by the introduced weighting and heuristics. Since DBpedia contains a lot of trivia facts, the exposure of even this result shows the potential of our approach. Especially since the above computation took just $0.68s$ there is definitely margin for more complex logic should the use case require or tolerate it.

Hitrate. The hitrate of our algorithm is above 95% which is high, considering the relatively small number of resources that had to actually be checked compared to the size of the entire dataset (10.8M resources). Checking a resource means retrieving the resource from the index and identifying the linked resources. This is under 6000 in most of the cases as shown in Figure 6.5. These results indicate that popular concepts on DBpedia are well interlinked and form a dense graph. Our optimization, with PageRank to reduce the rank of the adjacency matrix, does not eliminate many possible results.

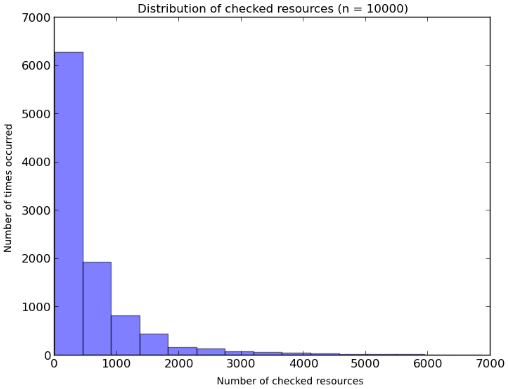


Figure 6.5: More than half the paths required less than 500 resources to check.

Path length. The mean length of the calculated paths is about 4 hops. The mean of the path length values is $\mu = 4.1$. The sudden dip in frequency for paths with length 4 is due to the test set with a random choice of starting points and destinations. The majority of these resources were geographical and are thus by nature linked with fewer steps than we would averagely expect. For example, the majority of cities and countries if often linked in two to three steps through some person who was born in one country and lived in a another country. It is unclear if this behavior would occur in other datasets. Typical to DBpedia is that a limited number of properties can be very common to many others. It is likely that if another dataset is structured hierarchical, contains mainly connected trees, or has a certain type of resource which has in comparison the entire size of the dataset a small amount of resources but links to many other entities (such as countries or cities).

Nevertheless, the distribution of the path lengths approximates a gamma function with $\mu = 3.4$ (see Figure 6.6). A phylogenetic tree or evolutionary tree shows the relationships among various (biological) entities based upon similarities between their characteristics. Our heuristic, the Jaccard, takes into account the similarity between resources' predicates (equivalent to characteristics) for finding a link between two resources. Our algorithm finds paths among a combination of two trees which are in structure similar to a phylogenetic tree. One tree which has as root the source and the other tree which has the destination as root. Numerical findings from Mir et al. confirm that the distribution of the distance, or path length, between two nodes in a phylogenetic tree equiprobably chosen, approximates a gamma distribution [35]. The probability to find a path is very low from a certain path length. Because of the gamma distribution we safely state that this justifies the choice of a termination of the algorithm after a fixed amount of steps. Most of the path lengths are centered around the statistical centralities the lower and upper boundary of the statistical mean and the mean of the gamma distribution.

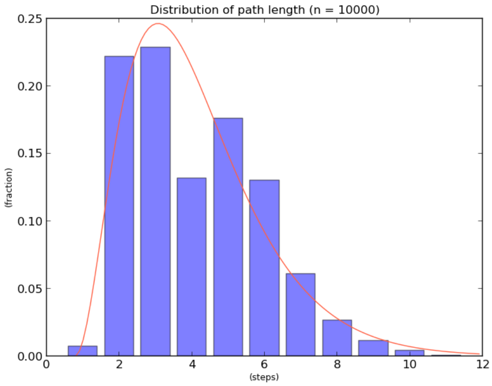


Figure 6.6: Normalized distribution of found path lengths has a peak of 3 near $\mu = 3.4$ of the fitted gamma distribution.

Execution time. The time complexity of A* depends on the complexity for the evaluation of the heuristic. The evaluation of our heuristic, the Jaccard, is linear to the number of predicates for the resources. Using our optimization we retained the linear execution time and have results in the cases in which we found a path despite the optimizations. We have approximated a scatterplot in Figure 6.7 with a linear curve. A* is guaranteed to find a path if the resources are connected. However with our optimization this is no longer the case, because the optimization limits the search space of the graph, it is thus possible that existing paths between resources are being left out of the search domain. Our algorithm, based on A*, was implemented for an optimal amount of resources to be checked. The result has the advantage that the number of resources to be checked grows linear but the disadvantage that there is no guarantee to always find a path that exists.

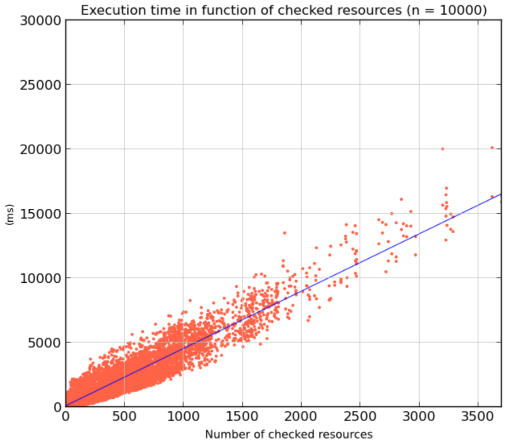


Figure 6.7: Execution time (y) is approximately a linear function of the checked resources (x) $y \approx 4.4x + k$

We can find most of the paths within an tolerable amount of time, a tolerable time for users when retrieving information is maximum 2 seconds [37]. The algorithm finds 60% of the paths within 2 seconds. With a notification to the user the tolerable time could extend to 10 seconds or even more. In 10 seconds we find a path for more than 95% of the queries.

We notice furthermore in Figure 6.8 that there is a linear relation in logarithmic space between execution time and path length. The results for longest paths with length 11 and 12 (excluded from the plots) are not relevant as they do no not occur frequently enough compared to the others. There is almost no difference between a path of length 1 and length 2 because other side-effects such as set-up time have an impact when the total execution time is in the order of $20 \sim 50ms$. This is what we could expect as the number of resources to check increases exponentially with increasing path length, see Figure 6.9. The use of an optimal index that ensures a constant retrieval time is crucial as the number of resources to check increases exponentially with increasing path length. The execution time is linear compared to the number of checked resources. This is ensured because the time to retrieve

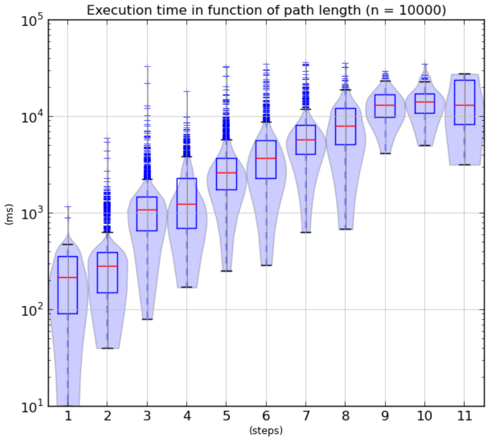


Figure 6.8: The execution time in function of path length appears to be linear in a logarithmic scale.

resources is also linear if the time to retrieve each resource from the index is always constant.

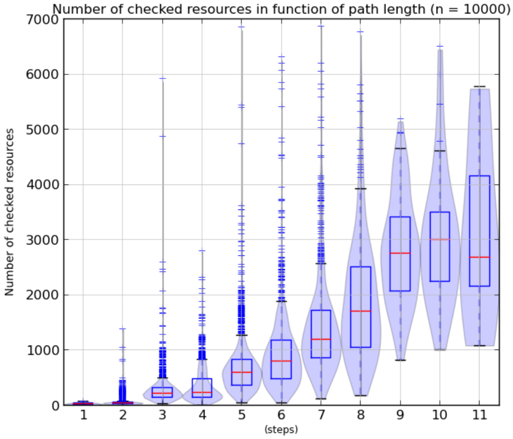


Figure 6.9: The number of checked resources grows exponentially with path length except when the amount of test queries is not high enough to draw a conclusion (path length 10 or 11).

Optimized Algorithm

To determine if the arbitrariness of a story is reduced, we validated that our optimization improved the link estimation between concepts mentioned in a story. To this end, we computed stories about the four highest ranked DBpedia scientists, according to their PageRank score[7]. Resources with a high PageRank are typically very well connected and have a high probability to lead to many arbitrary paths.

Initial Sample. We have determined the pairwise semantic relatedness of the story about them by applying the Normalized Google Distance (NGD). The results are shown in Table 6.6.

Table 6.6: The comparison between the base and optimized algorithm shows that the semantic relatedness can be improved in all cases except for the last two when the entities were already closely related, their NGD in the base algorithm was already relatively low.

No.	Query Context	Base Algorithm	NGD	Optimized Algorithm	NGD
S1	C._Linnaeus - C._Darwin	C._H._Merriam	0.50	J._W._Von_Goethe	**0.43**
S2	C._Linnaeus - A._Einstein	Aristotle	0.70	J._W._Von_Goethe	**0.45**
S3	C._Linnaeus - I._Newton	P._L._Maupertuis	0.48	D._Diderot	**0.40**
S4	A._Einstein - I._Newton	Physics	0.62	D._Hume	**0.45**
S5	C._Darwin - I._Newton	D._Hume	**0.38**	Royal_Liberty_School	0.40
S6	C._Darwin - A._Einstein	D._Hume	**0.43**	B._Spinoza	0.44

Table 6.6 shows that the entities *Aristotle* and *Physics* are included in the story when applying the original algorithm. These entities are perfect examples of *arbitrary* resources in a story which decreases the consistency. Except that they are related to science, it is unclear to the user why the algorithm 'reasoned' them to be in the story. When utilizing the optimized algorithm these entities are replaced by *J._W._Von_Goethe* and *D._Hume*.

Detailed Sample. To verify the results, we include the total semantic similarity of a path by computing the semantic relatedness between all neighboring node pairs in that path. As can be seen in Table 6.6, the optimized algorithm seemed to be able to improve the link estimation of the resulting paths. To evaluate the results we used three different similarity measures: W2V [8], NGD [8], and SemRank [2][3].

We used an online available Wiki2VecCorpus using vectors with dimension 1000, no stemming and 10skipgrams[9]. We computed the similarities based on that model

[7] http://people.aifb.kit.edu/ath#DBpedia_PageRank

[8] https://code.google.com/p/word2vec/

[9] https://github.com/idio/wiki2vec

Table 6.7: Abbreviations explained and short interpretation of the measures used.

Abbreviation	Description
W2Vs	Word2Vector similarity using Wikipedia English Corpus
NGD	Normalized Web Search Distance using Bing API
SR-C	SemRank - Conventional - No particular role for serendipity
SR-M	SemRank - Mixed - Serendipity plays partly a role
SR-D	SemRank - Discovery - Serendipity has a major role
PR	PageRank - Centrality Degree of a Node

by using *gensim*[10]. We implemented the NGD - generalized as the normalized web search distance, on top of the Bing Search API, using the same formula as depicted in the heuristic for the algorithm.

We applied SemRank to evaluate the paths, in particular to capture the serendipity of each path. The serendipity is measured by using a factor μ to indicate the so called 'refraction' how different each new step in a path is compared to the previous averaged over the entire path. Furthermore the information gain is modulated using the same factor μ. The information gain is computed from the weakest point along the path and an average of the rest. So that we get as formula for SemRank and a path p:

$$\text{SemRank}(\mu, p) = [\frac{1-\mu}{I(p)} + \mu I(p)] \times [1 + \mu R(p)], \tag{6.7}$$

where $I(p)$ is the overall information gain in the path and $R(p)$ is the average refraction. There are three special cases [3]: (i) **conventional** with $\mu = 0$ leading to $\text{SemRank}(0, p) = \frac{1}{I(p)}$, serendipity plays no role and so no emphasis is put on newly gained or unexpected information; (ii) **mixed** with $\mu = 0.5$ leading to $\text{SemRank}(0.5, p)$ $= [\frac{1}{2I(p)} + \frac{I(p)}{2}] \times [1 + \frac{R(p)}{2}]$, a balance between unexpected and newly gained information; and (iii) **discovery** with $\mu = 1$ leading to $\text{SemRank}(1, p) = I(p) \times [1 + R(p)]$, emphasizing unexpected and newly gained information.

The DBPedia PageRank[11] (PR) is an indicator for average 'hub' factor of resources and their neighbourhood based links, how 'common' they are [38].

Table 6.8 shows the various improvements of the control algorithm using different measures: both the base and optimized algorithms were configured with the same, the Jaccard distance, weight and heuristic.

Effect of Weights and Heuristics. The results, shown in Figure 6.10, confirm the findings in the detailed sample, but this time the base algorithm uses a combination

[10] `https://radimrehurek.com/gensim/`

[11] `http://people.aifb.kit.edu/ath#DBpedia_PageRank`

Table 6.8: Detailed comparison between the base and optimized algorithm.

	Measure	Higher Better?	S1	S2	S3	S4	S5	S6	AVG	STDEV
Base	SR-C	+	6.46	6.70	5.48	9.47	6.50	9.00	7.17	1.59
	SR-M	+	4.04	4.05	3.34	5.25	4.11	5.21	4.35	0.75
	SR-D	+	0.22	0.20	0.25	0.13	0.23	0.14	0.20	0.05
	NGD	-	0.64	0.69	0.48	0.31	0.48	0.29	0.48	0.16
	W2Vs	+	?	?	0.18	0.32	0.21	0.39	0.20	0.02
	PR	-	*2631.89*	66.27	179.50	62.39	357.36	62.39	166.38	128.58
Improved	SR-C	+	9.19	8.00	7.17	6.74	9.47	6.50	7.78	1.15
	SR-M	+	5.39	4.70	4.00	3.98	5.44	3.95	4.52	0.65
	SR-D	+	0.14	0.16	0.17	0.19	0.13	0.21	0.17	0.03
	NGD	-	0.53	0.22	0.60	0.38	0.32	0.55	0.45	0.14
	W2Vs	+	0.21	0.19	0.20	?	0.34	?	0.27	0.10
	PR	-	40.42	97.11	29.29	0.59	62.39	0.89	33.25	34.08

of the Combined Node Degree (CND) and the Jaccard distance, while the optimized algorithm was configured using a variety of heuristics and weights. To be able to compare the results with each other each of the SR measures are normalized as follows: $\text{SRn} = \frac{\text{SR}}{\max(\text{SR})}$.

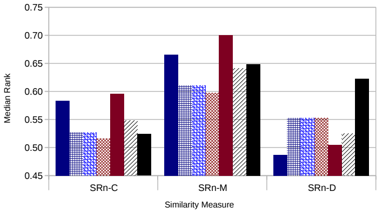


Figure 6.10: Effects of the different combinations of weights and heuristics on the measured SemRank.

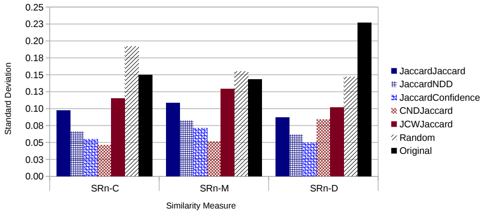


Figure 6.11: Standard deviation of the measured SemRank when using different heuristics.

The standard deviation of the results, shown in Figure 6.11, highly differs for each case. In particular when using a random number instead of a weighting function and a heuristic, leads to a high standard deviation, which is expected - given the randomness. The deviation is also relatively high when using the Jiang-Conrath distance as weight (JCW) and when using the base algorithm.

On the one hand the conventional and mixed mode for SemRank put less emphasis on novelty and focuses mainly on semantic association and information content. The jaccard distance combination used as weight and heuristic is not entirely surprisingly the best choice for this scenario. On the other hand the results of the improved algorithm with the common node degree confirm the results of the base algorithm with the common node degree as weight and the jaccard distance as heuristic is.

There is however a slightly lower rank when using the improved algorithm. Using the JCW however leads to even higher ranks. In terms of discovery, the base algorithm outperforms the JaccardJaccard combination. The CNDJaccard improved algorithm is able to slightly outperform all the other combinations.

User Judgments. We presented the output of each of the algorithms as a list of story facts using the scientists example cases S1 - S6 as shown in Table 6.8. Typically 1 up to 20 facts depending on the heuristics that were used. As with SemRank, we are interested in the serendipity as a balance between unexpected facts and relevant facts. We asked the users to rate the list of facts in terms of: (i) relevance; (ii) consistency; and (iii) discovery. The users had to indicate how well the list of facts scored according to them on a Likert scale from -2 (None, Not, Very Poor) to +2 (Most, Very, Very Good). A score of 0 (neutral) was only possible in the case of relevance. In total we collected 840 judgments, 20 judgments for each combination of scenario and heuristic. The overall results of the user judgments, rescaled to a score between 0 and 1 are: **relevancy** 0.45; **consistency** 0.45; and **discovery** 0.33. The scores around 0.5 can be interpreted as a disagreement between the users. The median standard deviations are 0.29; 0.31, and 0.30 respectively.

The overall score is below 0.5, this indicates that the majority of users judges most of the presented list of story facts below normal or expected relevancy, consistency and with little unexpected new facts. The standard deviation of the user judgments is relatively high, which means that they cover a broad range of judgments, i.e., some users are very positive while other users are very negative. The mixed results are likely due to varying expectations: some might expected more in-depth results while others appreciated the basic facts about the scientists. The suggested stories that center around a certain via-fact are not always considered relevant by some users even though the algorithms might consider them so. Some examples:

- The users least agreed on the relevancy of the following facts about Carl Linnaeus and Albert Einstein, a relevance score of **0.48** and a standard deviation of **0.31** when using the JCWJaccard:

  ```
  Carl Linnaeus and Baruch Spinoza are Expert, Intellectual and Scholar
  Baruch Spinoza's and Albert Einstein's are both Pantheists
  Intellectuals and Jewish Philosophers
  ```

- The second most relevant *and* consistent facts were found between Charles Darwin and Carl Linnaeus: a score of **0.65** and **0.6** respectively with CNDJaccard.

```
Copley Medal is the award of Alfred Russel Wallace and Charles Darwin
Alfred Russel Wallace's and Charles Darwin's awards are Royal Medal
and Copley Medal
Alfred Russel Wallace and Charles Darwin are known for their Natural
selection
Carl Linnaeus and Alfred Russel Wallace have as subject 'Fellows of
the Royal Society'
Carl Linnaeus and Alfred Russel Wallace are Biologists and Colleagues
```

- In terms of relevance the highest score also has the most agreement among users, generated by the original algorithm: a score of **0.8** and standard deviation **0.26**.

```
Albert Einstein's and Isaac Newton's field is Physics.
```

- In terms of discovery the highest score has relatively little agreement among users: **0.48** and standard deviation **0.30** with JCWJaccard:

```
Albert Einstein's and Charles Darwin's reward is Copley Medal.
```

The scores for relevancy, consistency, and discovery as unexpected - but relevant - facts are highly dependent on the user who judges. Some users might be interested in the more trivial or arbitrary path as well. Nevertheless, the overall judgment served as a baseline to compare the judgments with the same combinations of heuristics and weights as before.

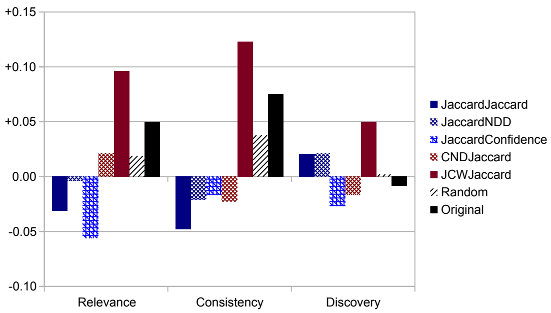


Figure 6.12: The effect of the heuristics according to user judgments compared to the overall median. The JCWJaccard confirms already good results with SemRank. The CNDJaccard scores relatively well in terms of relevance.

The most consistent output was generated with the Jaccard distance used both as weight and heuristic; or as heuristic in combination with the Jiang-Conrath distance as weight. The most arbitrary facts occur in a story when using the combined node degree as weight with the Jaccard distance as heuristic, both in the optimized and the base algorithm. User judgments confirm the findings for the Jiang-Conrath weight and the base algorithm and for the Jaccard distance used as weight and heuristic in terms of discovery. There is no clear positive effect however according the users in terms of consistency and relevancy there.

Summary

A technique combining pre-processing and indexing of datasets is used to implement a base algorithm for finding paths between two resources in large datasets within a couple of seconds. Using linked data in combination with a specialized search index enabled pathfinding algorithms to work in large linked datasets within a tolerable time for users. The base algorithm delivers a graph-based search approach to explore the connectivity of resources. A major contribution is the minimization of the size of the candidate pool of nodes to tweak execution performance and to increase the quality of the resulting paths. To do this, different ranking algorithms (PageRank, HITS, SVD) were compared before finally applying the PageRank algorithm. The testcase using the DBpedia dataset showed promising performance results, but also exposed issues that some paths were too arbitrary. An optimization of the base algorithm improves the serendipity level of the relations and mitigates arbitrariness by increasing the relevance of links between nodes through additional pre-selection and refinement steps. In both cases, composing stories rely on finding indirect relationships in linked data based on A* path search. Furthermore the storytelling algorithms were tested with several heuristics and weights. The results made clear that the choice of heuristics and weight requires careful consideration, especially as they clearly have a big impact on the result set.

References

[1] A. Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[2] B. Aleman-Meza, C. Halaschek-Weiner, I. B. Arpinar, C. Ramakrishnan, and A. P. Sheth. Ranking complex relationships on the semantic web. *Internet Computing, IEEE*, 9(3):37–44, 2005.

[3] K. Anyanwu, A. Maduko, and A. Sheth. Semrank: ranking complex relationship search results on the semantic web. *Proceedings of the 14th international conference on World Wide Web*. WWW '05, pages 117–127, ACM, Chiba, Japan, 2005.

[4] R. Bellman. On a routing problem. *Quart. Appl. Math.* 16:87–90, 1958.

[5] C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.

[6] B. Chen, X. Dong, D. Jiao, H. Wang, Q. Zhu, Y. Ding, and D. J. Wild. Chem2bio2rdf: a semantic framework for linking and data mining chemogenomic and systems chemical biology data. *BMC bioinformatics*, 11(1):1, 2010.

[7] B. Cherkassky, A. Goldberg, and T. Radzik. Shortest paths algorithms: theory and experimental evaluation. *Mathematical programming*, 73(2):129–174, 1996.

[8] R. L. Cilibrasi and P. M. Vitanyi. The google similarity distance. *Knowledge and Data Engineering, IEEE Transactions on*, 19(3):370–383, 2007.

[9] X. Cui and H. Shi. Astar-based pathfinding in modern computer games. *International Journal of Computer Science and Network Security*, 11(1):125–130, 2011.

[10] B. De Meester, T. De Nies, L. De Vocht, R. Verborgh, E. Mannens, and R. Van de Walle. StoryBlink: a semantic web approach for linking stories. eng. *Proceedings of the 14th International Semantic Web Conference (ISWC) Posters & Demonstrations Track*, pages 4, Ceur, Bethlehem, Pennsylvania, USA, 2015.

[11] T. De Nies, C. Beecks, F. Godin, W. D. Neve, G. Stepien, D. Arndt, L. De Vocht, R. Verborgh, T. Seidl, E. Mannens, and R. V. de Walle. A distance-based approach for semantic dissimilarity in knowledge graphs. *Proceedings of the 10th International Conference on Semantic Computing*. accepted, 2016.

[12] L. De Vocht, C. Beecks, R. Verborgh, E. Mannens, T. Seidl, and R. Van de Walle. Effect of heuristics on serendipity in path-based storytelling with linked data. S. Yamamoto, editors, *Human Interface and the Management of Information:Information, Design and Interaction: 18th International Conference, HCI International 2016 Toronto, Canada, July 17-22, 2016, Proceedings, Part I*, pages 238–251, Springer International Publishing, Cham, 2016.

[13] L. De Vocht, C. Beecks, R. Verborgh, T. Seidl, E. Mannens, and R. Van de Walle. Improving semantic relatedness in paths for storytelling with linked data on the web. *The Semantic Web: ESWC 2015 Satellite Events*, pages 31–35, 2015.

[14] L. De Vocht, S. Coppens, R. Verborgh, M. Vander Sande, E. Mannens, and R. Van de Walle. Discovering meaningful connections between resources in the web of data. *Proceedings of the 6th Workshop on Linked Data on the Web (LDOW2013)*, Rio de Janeiro, Brazil, 2013.

[15] L. De Vocht, R. Verborgh, and E. Mannens. Using triple pattern fragments to enable streaming of top-k shortest paths via the web. In. *Semantic Web Challenges: Third SemWebEval Challenge at ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Revised Selected Papers*. Ed. by H. Sack, S. Dietze, A. Tordai, and C. Lange. Springer International Publishing, Cham, 2016, pages 228–240.

[16] R. Delbru, S. Campinas, and G. Tummarello. Searching web data: an entity retrieval and high-performance indexing model. *Web Semantics: Science, Services and Agents on the World Wide Web*, 10:33–58, 2012.

[17] L. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[18] E. Dijkstra. A note on two problems in connexion with graphs. English. *Numerische Mathematik*, 11:269–271, 1959.

[19] C. H. Q. Ding, X. He, P. Husbands, H. Zha, and H. D. Simon. PageRank: hits and a unified framework for link analysis. *SDM*, 2003.

[20] T. Eliassi-rad and E. Chow. Using ontological information to accelerate path-finding in large semantic graphs: a probabilistic approach. *Proceedings of the 29th national conference on Artificial Intelligence*, 2005.

[21] A. Foster and N. Ford. Serendipity and information seeking: an empirical study. *Journal of Documentation*, 59(3):321–340, 2003.

[22] V. Franzoni, M. Mencacci, P. Mengoni, and A. Milani. Heuristics for semantic path search in wikipedia. *Computational Science and Its Applications–ICCSA*, pages 327–340, 2014.

[23] F. Godin, T. De Nies, C. Beecks, L. De Vocht, W. De Neve, E. Mannens, T. Seidl, and R. Van de Walle. The normalized freebase distance. eng. In: V. Presutti, E. Blomqvist, R. Troncy, H. Sack, I. Papadakis, and A. Tordai, editors, *The Semantic Web: ESWC 2014 Satellite Events*. Volume 8798 of pages 218–221. Springer, Anissaras, Greece, 2014.

[24] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100–107, 1968.

[25] B. Haslhofer, E. Momeni Roochi, B. Schandl, and S. Zander. Europeana RDF store report. Tech. rep. University of Vienna, 2011.

[26] B. He, J. Tang, Y. Ding, H. Wang, Y. Sun, J. H. Shin, B. Chen, G. Moorthy, J. Qiu, P. Desai, and D. J. Wild. Mining relational paths in integrated biomedical data. *PLoS ONE*, 6(12):e27506, December 2011.

[27] P. Heim, S. Lohmann, and T. Stegemann. Interactive relationship discovery via the semantic web. In: *The Semantic Web: Research and Applications*, pages 303–317. Springer, 2010.

[28] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *Proceedings of the 10th International Conference on Research in Computational Linguistics,* 1997.

[29] S. Kulkarni and D. Caragea. Computation of the semantic relatedness between words using concept clouds. *KDIR*, pages 183–188, 2009.

[30] D. Kumar, N. Ramakrishnan, R. F. Helm, and M. Potts. Algorithms for storytelling. *Knowledge and Data Engineering, IEEE Transactions on*, 20(6):736–751, 2008.

[31] P. Lalanda. Two complementary patterns to build multi-expert systems. *Pattern Languages of Programs*, 1997.

[32] G. Laporte. The vehicle routing problem: an overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992.

[33] S. Maharajan. Performance of native SPARQL query processors. MA thesis. Uppsala University, 2012.

[34] L. Mazuel and N. Sabouret. Semantic relatedness measure using object properties in an ontology. Springer, 2008.

[35] A. Mir and F. Rosselló. On the distribution of the distances between pairs of leaves in phylogenetic trees. *BIOTECHNO 2011, The Third International Conference on Bioinformatics, Biocomputational Systems and Biotechnologies*, pages 100–103, 2011.

[36] J. L. Moore, F. Steinke, and V. Tresp. A novel metric for information retrieval in semantic networks. *The Semantic Web: ESWC 2011 Workshops*. Springer, pages 65–79, 2012.

[37] F. F.-H. Nah. A study on tolerable waiting time: how long are web users willing to wait? *Behaviour & Information Technology*, 23(3):153–163, 2004.

[38] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: bringing order to the web. Technical Report 1999-66. Previous number = SIDL-WP-1999-0120. Stanford InfoLab, November 1999.

[39] J. L. Redondo Garcia, L. De Vocht, R. Troncy, E. Mannens, and R. Van de Walle. Describing and contextualizing events in tv news show. *Proceedings of the 23rd International Conference on World Wide Web*. WWW '14 Companion, pages 759–764, ACM, Seoul, Korea, 2014.

[40] M. Vander Sande, R. Verborgh, S. Coppens, T. De Nies, P. Debevere, L. De Vocht, P. De Potter, D. Van Deursen, E. Mannens, and R. Van de Walle. Everything is connected: using Linked Data for multimedia narration of connections between concepts. B. Glimm and D. Huynh, editors, *Proceedings of the 11th International Semantic Web Conference Posters and Demo Track*. Volume 914 of CEUR Workshop Proceedings, CEUR-WS.org, 2012.

[41] R. Verborgh, M. Vander Sande, O. Hartig, J. Van Herwegen, L. De Vocht, B. De Meester, G. Haesendonck, and P. Colpaert. Triple Pattern Fragments: a low-cost knowledge graph interface for the Web. eng. *Web Semantics*, 37-38:184–206, 2016.

Part III

Use Case

Chapter 7

Web 2.0 for Scientific Research

Sou discipulo que aprende, [I am a student who learns,]
Sou mestre que da lição. [I am a master who teaches.]

—Traditional Capoeira Song.

The Web enables new ways to share and explore research. Research collaboration platforms like Mendeley or ResearchGate are an example of this. Faceted search, keyword matching and filtering are the main techniques used in current search interfaces. They focus mostly on narrowing down the search scope. This chapter explains a use case for visualizing linked data of research-related data sources to address the interactive aspect of exploring relationships between resources. By visualizing links between conferences, publications and proceedings users can discover relationships.

7.1 Introduction

Peer-reviewed research publications as well as related metadata from bibliography archives are widely available on the web. They offer a vast amount of information on related publications and can facilitate suggesting new contacts, collaborators, and interesting custom events. Usually the platforms supporting this information exchange expose a Web API that allows access to the structured content, or the information is present as Linked Data. Facilitating search in an interlinked repository of linked datasets for research environments is useful because it is still a laborious task for researchers to construct separate search queries for each of those services.

The enrichment with Linked Data resources allows researchers to find a vast amount of resources implicitly related to them.

The Linked Open Data (LOD) Cloud has reached a respectable size and publication repositories are very numerous. Around 10% of the overall distribution of triples comes from research publication repositories and publications are the source of around 30% of the overall links distribution[1]. Information present within the LOD Cloud offers a solid base of re-usable information to weave the Web and adapt information for researchers and scientists. The usage of such systems with linked data is getting wide spread nowadays in a variety of topical domains [20].

Definition

Research 2.0 depicts using Web – 2.0 – tools and principles in scientific research and learning. It is an application field of "Technology Enhanced Learning" which covers the entirety of learning and research with use of new media. It is an approach to science that maximally leverages information-sharing and collaboration tools and emphasizes the advantages of increased online collaboration between researchers. Researchers often use Social media, such as Twitter and Facebook, during scientific events to comment and discuss about each other's work, and to exchange research related materials [11]. They also use Web collaboration tools like Mendeley or ResearchGate to exchange their scientific work. Such academic social networks have become wide-spread and can have millions of regular users [26]. These tools and services have APIs, publishing feeds, and specially designed interfaces based on social profiles [18, 25]. These tools and services are in line with the principles of Research 2.0 [25].

Purpose

The purpose of this use case is to offer a set of tools and services which researchers can use to discover resources as well as to facilitate collaboration via the web. The goal is to present users how they are (indirectly) related based on their institutions, visited locations, and conferences they contributed to.

One of the key variables is the end-user usability of:

- semantically enriched researcher profiles;

[1] http://lod-cloud.net/state/

- relations between researchers based upon the semantic analysis of researcher's tweets and aligned with information about conferences and proceedings.

As a measure of usability we investigated the ability to support the construction of a good cognitive model of the underlying data and the relations within the data. Finally, we measured the effectiveness and productivity of the interface by checking to which extent end-users carry out knowledge-intensive and analytical tasks. This use case on a personalized interactive exploratory search environment follows the architectural model of the techniques explained in chapters 4, 5, and 6 with data from several open Linked Data repositories including scientific publication archives and social media.

Research Questions

This use case about researchers exploring information to gain insight in the people, conferences, or publications they wanted to find out more about, led to following questions:

- How does this approach compare to other related approaches in terms of user actions and precision?
- How well does the interactive approach perform in scenarios focused on more straightforward, keyword-based search tasks.
- When does this approach excel in revealing relationships compared to state of the art?

We implemented the prototype of a 'research exploration tool', called ResXplorer [2] to test the use case. Chapter 8 goes into more details about the tool.

7.2 Background

The evolution of the Web 2.0 enabled many users via wikis, blogs and other content publishing platforms to become the main content providers on the web. The data is available under the form of raw data, posts, threads, tags, and user information mappable to semantic form, since widely used and accepted vocabularies for many

[2] http://www.resxplorer.org

domains exist. However, the mass produced data remains in so-called ‘data silos’ bound to a specific platform or somewhere within databases. The access to these data sources is associated with specialized application interfaces (API’s) which requires specialized technical knowledge to retrieve the data in a desirable form. Many information public interest sources remain captured behind a so-called ‘walled garden’. Combining information resources over the walls leads to a high degree of mismatches between vocabulary and data structure of the different sources [13]. When users formulate a (Web) search in a certain context across multiple data sources, it often includes keywords. In many cases the semantic importance and meaning of the keyword is not considered. The keyword order and combination in a query affects the context, the precise goal of the search and thus the results.

Researchers’ use of Social Media

As the number of Web 2.0 users increased, Social Media arrived, commonly known as Social Networks. Researchers especially appreciate this development. For instance, studies on the use of microblogs like Twitter[3] [9, 10] within the science community showed that researchers were using Twitter to discuss and asynchronously communicate on topics during conferences [19] and in their everyday work [16, 19]. A survey of the use of Twitter for scientific purposes [16] has shown that Twitter is not only a communication medium but also reliable source of data for scientific analysis, profiling tasks, and trends detection [1, 17, 21, 23]. Twitter hashtags have a strong influence on the structuring of communication within Twitter as well as for community building [1, 15].

The Web 2.0 in Scientific Research

The Web 2.0 for Science, also known as *Science 2.0* or *Research 2.0* aims to adapt the Web 2.0 to the needs of researchers. The purpose of our research is to offer a set of tools and services which researcher can use to discover resources, such as publications or events they might be interested in [7, 24]. These tools and services, according to the specifications of Research 2.0, are considered as mash-ups, API’s, publishing feeds, search and discovery service and specially designed interfaces based on social profiles [18, 25]. Research 2.0 comprises interacting with information published on Social Media, online collaboration platforms, and other Web 2.0 tools. Weaving

[3] http://www.twitter.com

microblogs in the Web of Data is interesting from the perspective of researcher centric semantic search. Twitter, as exemplary microblog Social Media platform, can help resolving scientific citations [28].

On top of that, most research publications are available via the Web, as most of the digital libraries and scientific online journals offer access to their content. Usually they need a paid membership to get full access to their articles, but most of the educational institutions can afford this kind of service. At the same time a growing number of "Open Journals" offer free access to all published works. Most prominent archives in this area are Directory Of Open Access Journals (DOAJ)[4] as well as Online Journals[5]. The efforts to make the scientific resource sharing a reality concerns the researchers in science and educational informational systems for a long time. The products of such quests lead to an increasing variety of heterogeneous technologies, schema, repositories and query mechanisms. Since Linked Data emerged and the Semantic Web evolved aiming at Web wide interoperability [3, 4], the problem of sharing resources is beginning to resolve and Linked Data found a wide acceptance within this community.

This trend produces a constantly growing amount of publicly available Linked Data about scientific repositories. Within the research community also commercial digital libraries like ACM (Association for Computer Machinery) Digital Library[6] started to publish their archives into the LOD Cloud [12] providing in this special case more than 12 million triples. Parallel to the commercial scientific content providers some academic institutions as well as most famous public libraries (Library of Congress[7], British National Library[8] and Bibliothèque Nationale de France[9]) provided their public Linked Data.

Linked Data-based Interfaces for Research Exploration

In the past there were attempts to visualize research networks but most of them did not rely on linked data. The below mentioned works based on research linked data consider visualizations as a supportive mean to the presented information.

[4] http://www.doaj.org/
[5] http://online-journals.org/
[6] `http://acm.rkbexplorer.com/`
[7] `http://id.loc.gov`
[8] `http://bnb.data.bl.uk`
[9] `http://data.bnf.fr`

The Semantic Web Journal published its own Drupal-based journal management system [14] focusing on providing a novel user interface. Among others, they provide graph-based research networks that visualize the emerging research networks as researchers author papers together or they review the different submissions. *RKB Explorer*[10] [12] is a visual browser which originated from the ReSIST[11] network of excellence, which unites within many sources of scientific data. This visual browsing interface is based on categorised pre-selection and focuses on people, organisations, publications, and courses and materials. The search always centers around the selected category which makes the context based browsing less flexible but focused. Within the visualisation RKB Explorer evaluates relations of the first degree. In comparison to RKB Explorer our approach is more user and search centric rather than concept and context centric. In our interface, a user profile affects the pre-selection of search results. Users can configure the search context by executing searches for resources or by expanding one or more resources. *BibBase*[12] [29] has an interface to leverage the personal publications into the Web of Data and integrates the retrieval of author publications with a small sample from Mendeley[13], DBLP[14], and Zotero[15]. Finally, "TalkExplorer" [27] takes into consideration bookmarks and tags for the visualizations of the research groups and puts the focus on providing recommendations rather than exploring the underlying dataset. In our workflow we make abstraction of the query creation process and use pre-defined query templates to facilitate the creation of the visualizations.

Related Work

The coverage of user driven search evaluations aspects which consider the visual representation and analysis of search results and interaction possibilities is important. The implementation of this use case shares the goal of search, data about research publications, and intended audience with Google Scholar (GScholar)[16]; Microsoft

[10] http://www.rkbexplorer.com
[11] http://www.resist-noe.org/
[12] http://bibbase.org
[13] http://www.mendeley.com/
[14] http://www.informatik.uni-trier.de/~ley/db/
[15] http://zotero.org
[16] http://scholar.google.com

Academic (MA) Search[17]; ARnet Miner [22][18]; Falcons [5][19]; and Faceted DBLP Search[20].

There is a spread between visually more advanced solutions like MA Search and ARNet Miner and those with less search interface interactivity possibilities like Google Scholar, Faceted DBLP, and Falcons. To outline the differences between conventional search interfaces for scientific resources and the implemented approach, we used a set of "Visual representation and analytics" based on guidelines identified by [6]. Table 7.1 compares the features of the search interfaces used in the expert evaluation. Industry references as MA Search and Google Scholar lack the interactivity with a visual representation, although MA Search for instance offers visual interfaces to the search results. On the other hand, ARNet Miner supports various visualizations based on data mining algorithms like, e.g., clustering, executed on the retrieved data in combination with the search results. ArnetMiner distinguishes between the networks (star graph of co-authors) and the communities of researchers (simple graphs). Falcons Object Search [5] is considered as a keyword-based search engine for linked objects with extensive virtual documents indexed. Those documents consist from associated literals but also from the textual descriptions of associated links and linked objects. The results are ranked according to a combination of their relevance to the query and their popularity. Falcons allows enhanced text based browsing of Linked Data as well as filtering on concepts and relations besides a classical list representation. Faceted DBLP features an interactive, all-round faceted search interface. The search approach in this case resides on DBLP++ data which enhances DBLP with additional keywords and abstracts as available on public web pages. It integrates facets on Time, Venues, Publications Years, and Authors and delivers the results in various formats. These formats include: BibTeX, regular web pages, DOI identifiers, or RDF. Faceted DBLP offers a good flexibility in filtering and narrowing down the results as well as implementing basic syntactic query expansion based upon single word and whole phrase in an anonymous way. Retrieval is done by classic search engines and result selection is done by ranking without any possible relation to the user profile.

[17] http://academic.research.microsoft.com

[18] http://artnetminer.org

[19] http://ws.nju.edu.cn/falcons

[20] http://dblp.l3s.de/

Table 7.1: Comparision of functionality of different search interfaces for research.

Usability Criterion	ResXplorer	MA Search	GScholar	ARnet Miner	Falcons	Faceted DBLP
Query (forms / keyword)	●	●	●	●	●	●
Query (formal syntax)	◐	●	●	●	●	◐
View results as ordered list	○	●	●	●	●	●
Visual presentation	●	●	○	●	○	○
Interactively refine search	●	○	○	◐	●	◐
Combine and relate searches	●	○	○	○	○	○
Data overview	●	●	○	●	●	●
Detail on demand	●	●	●	●	●	●
Generic / Engine Reusable	●	?	?	◐	?	?
Support for scalability	●	●	●	●	●	●
Filtering	◐	●	●	●	●	●
History	●	◐	◐	○	○	○
View original source	●	●	●	◐	●	●
Feature coverage	●= full	◐= partially	○= none	? = uncertain		

7.3 Approach

When looking for the next practical piece of information or when trying to find a solution for a problem that requires out-of-the-box thinking (e.g., when forming the exact search query requires background knowledge of a domain unfamiliar to the researcher). The interaction diagram in Figure 7.1 shows how researchers explore research objects. The research objects are made available through a two layer abstraction consisting of the: (i) data model; and (ii) user model.

Researchers can define and select their *intended* search goal over several iterations. When users are looking for new leads, they get an overview of possible objects of interest (similar to points of interest on a street map) by having their activities and contributions linked on social media and other platforms such as their own research publications profile.

We will illustrate the points above with a running example and take a computer science researcher investigating the Web. During scientific conferences on the subject like *The World Wide Web* Conference the researcher is regularly posting on Twitter and using the conference hashtag. At some point the researcher might be interested in figuring out more about search algorithms, who is involved, and if there is any match with some publications the research participated as a co-author. The search starts with the researcher centralized in the middle and the researcher chooses the most relevant option based on the suggestions, this is shown in Figure 7.2 and 7.3.

Furthermore, this reveals a first relationship between the researcher and this particular publication, also depicted in Figure 7.4. Further actions from this point available

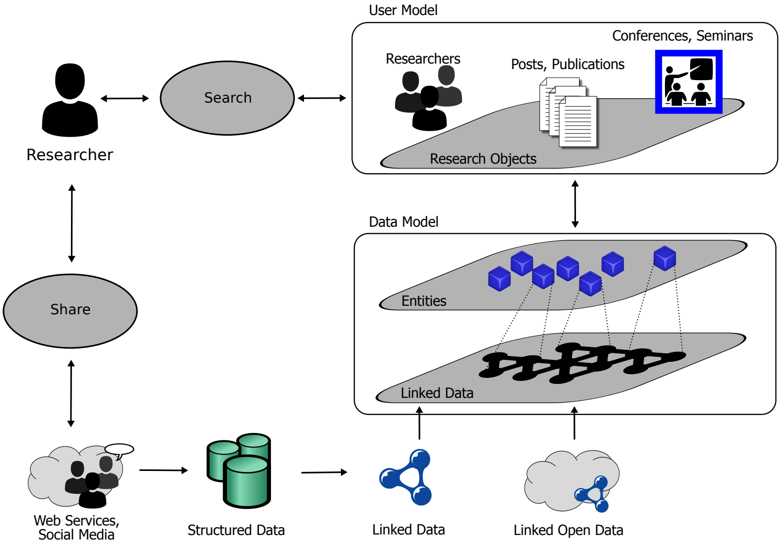


Figure 7.1: The information researchers share via the Web services of research collaboration tools and social media is structured and transformed to RDF and interlinked with Linked Open Data. The resulting entities in the data model form the base for the user model. This process is outlined in Section **7.3**. When researchers search, they interact indirectly with the user model which we detail in Section **7.3**.

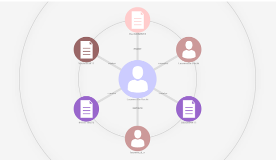
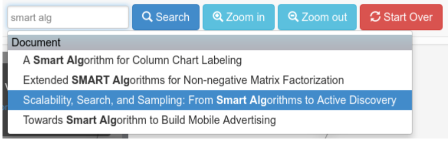


Figure 7.2: The search starts with the researcher centralized and some directly resources shown around.



Document

A **Smart Alg**orithm for Column Chart Labeling

Extended **SMART Alg**orithms for Non-negative Matrix Factorization

Scalability, Search, and Sampling: From **Smart Alg**orithms to Active Discovery

Towards **Smart Alg**orithm to Build Mobile Advertising

Figure 7.3: The researcher chooses the most related resource.

to the user are, searching again using other keywords or finding out more about a certain search result by clicking on them. This will center the search around the publication and give a new perspective.

More details are given given in Chapter 8 on how it is decided which exact resources are being revealed. Regardless of the resources revealed in this case, the figure shows documents, people and relationships between them. Furthermore, the visualization 'hides' that the searches are in data instead of documents. The only arguable clue that the search is in data, is the prominent graph structure of the visualization. This

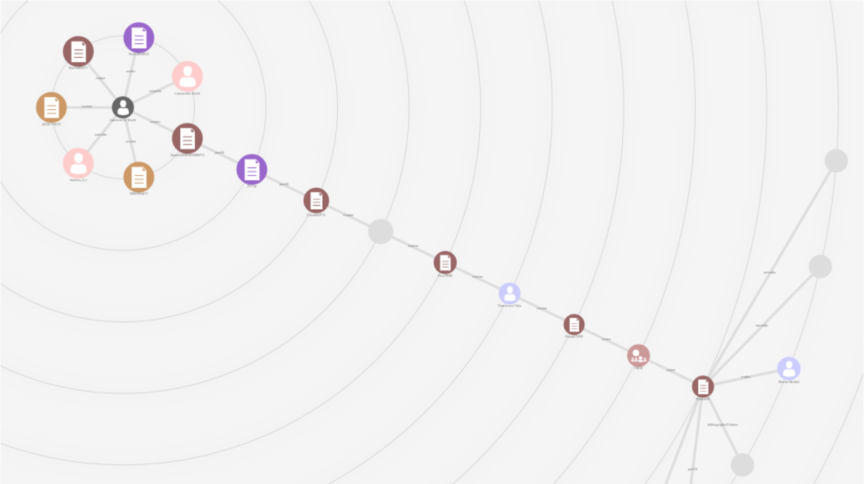


Figure 7.4: The revealed relationship between the researcher and the found publication.

is precisely due to the two-layered model of the data that is being searched and its representation as research objects for the user, as detailed in Figure 7.1.

User Model

Research Objects are a method to identify, aggregate, and exchange research data via the Web. They center and group refined entities of extracted and integrated data in the Data Model and represent [8]:

- Events: scientific conferences, seminars and/or lectures
- Publications: articles, reports, tutorials and/or posts
- Locations: both real-world and online (web pages, webinars)
- Concepts: topics, categories and/or classifications

Research objects enable and facilitate the use of research related information. The metadata that describes research objects facilitates searching and retrieving them.

Defining Research Objects. A single *Research Object* can contain links to and information about an online tutorial, details about a seminar, links to fragments of related papers and tutors or people who are known to have contributed to the entities of this specific object. Researchers define a search query for their research and have it parsed by our system for identification in terms of the User Model. The use of research objects in a user model should provide the reproducibility that enables validation of research results [2]. We align the entities present in the Data Model with the registered activities of researchers by providing their profiles and feeds of social media. Researchers generate those by sharing and monitoring online activities such as blogs, (micro)posts, tags, shares, and other resources.

Searching Research Objects. Searches center around several research targets that a researcher wants to relate with another. Searches also combine related resources based on common links they share, such as being related to and containing more information about a Research Object. The users generate their own views by exploring and searching among the Research Objects in the model and can share or compare those with other researchers or earlier searches. All those views together lead to a personalized environment. This will boost interaction with and grouping of similar views and objects to bigger packages that ultimately lead to the discovery of even more relations. The mapping of all objects for users are customarily based on their "researcher profile". Each researcher's profile is extracted based on the content researchers monitor on social media or the resources shared over it. Most of the researchers today own a profile in a scientific or common social network like Twitter and Facebook[21] or on research related platforms like ResearchGate[22], Mendeley, or Google Scholar.

Impact of Including Tweets Example. We mentioned in the running example that the researcher is active on Twitter during conferences related on the Web. When the researcher is exploring resources related to one of the *World Wide Web* conferences, the search system may be able to provide more information by taking into account social data. Figure 7.5 shows how the researcher tweets about a presentation on smart algorithms by Bob during one of the *World Wide Web* conferences and another researcher 'Anna', tweets about a paper that is published in the conference proceedings. They both use a hashtag that is commonly known in their scientific community to be associated with the conference *#WWW*. This leads to a direct connection between them. Without these tweets, a search would never be able to expose this connection between the two researchers. Anna is in fact a co-author of

[21] `http://www.facebook.com/`
[22] `http://www.researchgate.net/`

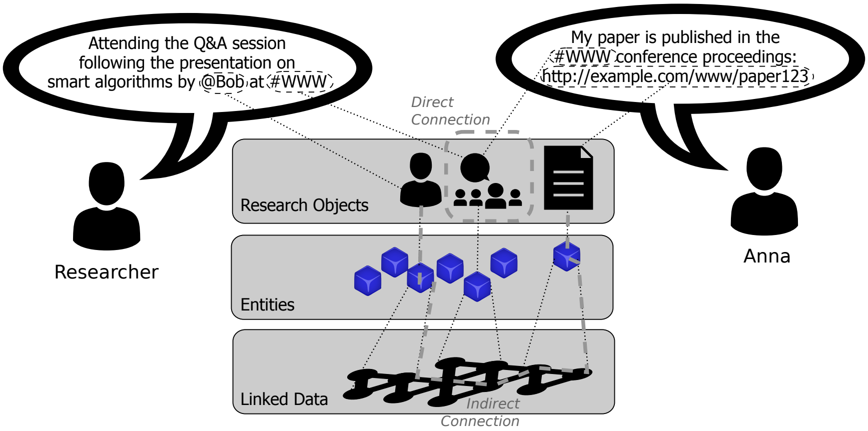

Attending the Q&A session following the presentation on smart algorithms by @Bob at #WWW
My paper is published in the #WWW conference proceedings: http://example.com/www/paper123
Direct Connection
Researcher
Anna
Research Objects
Entities
Linked Data
Indirect Connection

Figure 7.5: Conversations on social media contribute to exposing direct and indirect connections between resources.

Bob, information that cannot be derived from the tweets, but this may be derived from a digital archive, or a research collaboration platform. In any case, the mention *@Bob* by the researcher enables a search system to expose an additional, but indirect connection between the researcher and Anna. So there are at least two new potential relationships between resources that may potentially be revealed:

(i) **Direct**: The researcher and Anna both mentioned *#WWW*, the *World Wide Web Conference*.

(ii) **Indirect** The researcher mentions Bob, Bob is an author of the paper with URL *http://example.com/www/paper123*, and Anna is also a co-author of this paper.

The indirect connections are usually revealed because the representation of entities in the data model is suitable for this.

Data Model

The Data Model has two spaces. It has a Linked Data space and an Entity space. The former is the representation of the data loaded into the model and the latter are the entities, each having a URI, a label, a type and a description consisting of one or more Linked Data triples. In this section we describe the two types of data that we model: Research Data and Linked Data extracted from social media.

Research Data. Research data is described as Linked Data using state-of-the-art vocabularies, detailed in Section 8.2. We model research data with respect to their usage and wide popularity within the Semantic Web community, as well as to their applicability for the proposed use case. One of the modeling domains of interest are scientific events and their relatedness to bibliographical archives.

Linked Data from social media. We created an annotated set of extracted conference hashtags mentioned in tweets of researchers which would be associated with corresponding tweets and which can be used for further mining tasks like label based matching of scientific events in Linked Data sets, e.g., COLINDA, or DBLP. The motivation for linking data from social media: ‘social data’ as such, is threefold:

(i) **Link discovery** To allow detecting and creating links between the users and the data they are exploring.

(ii) **Timely context** To enforce a timely and personalized context to the search.

(iii) **Relationships** To add additional relationships between users and resources that are contained in the more static data and potentially introduce additional references to other Linked Open Data.

In this search use case, besides persons, locations, conferences and scientific publications, the researcher oneself is an important resource for the context.

Summary

The presented use case focuses on revealing relations between indirectly related resources about publications, conferences, and researchers. The domain of this use case became increasingly relevant due to the fact that during scientific conferences the use of social media, in particular microblogs became more important. The use case fits against a background on search interfaces, social media, and linked data on the subject. The approach for the use case entails a two-layered model focusing on the data and the semantics. In this use case, researchers explore so-called 'research objects' through this abstraction layer. The implementation and evaluation of this approach is outlined in the next chapter (8).

References

[1] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence on twitter. *Proceedings of the fourth ACM international conference on Web search and data mining*. WSDM ́11, pages 65–74, ACM, Hong Kong, China, 2011.

[2] S. Bechhofer, I. Buchan, D. D. Roure, P. Missier, J. Ainsworth, J. Bhagat, P. Couch, D. Cruickshank, M. Delderfield, I. Dunlop, M. Gamble, D. Michaelides, S. Owen, D. Newman, S. Sufi, and C. Goble. Why linked data is not enough for scientists. *Future Generation Computer Systems*, 29(2):599–611, 2013.

[3] C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee. Linked data on the web (ldow2008). *Proceedings of the 17th international conference on World Wide Web*. WWW '08, pages 1265–1266, ACM, Beijing, China, 2008.

[4] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - a crystallization point for the web of data. *Web Semantics*, 7(3):154–165, September 2009.

[5] G. Cheng and Y. Qu. Searching linked objects with falcons: approach, implementation and evaluation. *Int. J. Semantic Web Inf. Syst.* 5(3):49–70, 2009.

[6] A.-S. Dadzie and M. Rowe. Approaches to visualising Linked Data: a survey. *Semant. web*, 2(2): 89–124, April 2011.

[7] L. De Vocht, S. Softic, M. Ebner, and H. Mühlburger. Semantically driven social data aggregation interfaces for research 2.0. *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*. i-KNOW 2011, pages 43:1–43:9, ACM, Graz, Austria, 2011.

[8] L. De Vocht, D. Van Deursen, E. Mannens, and R. Van de Walle. A semantic approach to cross-disciplinary research collaboration. *Internation Journal of Emerging Technologies in Learning (iJET)*, 7(S2):22–30, 2012.

[9] M. Ebner, T. Altmann, and S. Softic. @twitter analysis of #edmedia10 - is the #informationstream usable for the #mass. *Form@re - Open Journal per la formazione in rete*, 11(74), 2011.

[10] M. Ebner, H. Mühlburger, S. Schaffert, M. Schiefner, W. Reinhardt, and S. Wheeler. Getting granular on twitter: tweets from a conference and their limited usefulness for non-participants. In: *Key Competencies in the Knowledge Society*. Volume 324 of IFIP Advances in Information and Communication Technology, pages 102–113. Springer Berlin Heidelberg, 2010.

[11] M. Ebner and W. Reinhardt. Social networking in scientific conferences - Twitter as tool for strengthen a scientific community. U. Cress, V. Dimitrova, and M. Specht, editors, *Learning in the Synergy of Multiple Disciplines, Proceedings of the EC-TEL 2009*. Volume 5794 of Lecture Notes in Computer Science, Springer, Berlin/Heidelberg, October 2009.

[12] H. Glaser, I. C. Millard, and A. Jaffri. Rkbexplorer.com: a knowledge driven infrastructure for linked data providers. *Proceedings of the 5th European semantic web conference on The semantic web: research and applications*. ESWC'08, pages 797–801, Springer-Verlag, Tenerife, Canary Islands, Spain, 2008.

[13] D. M. Herzig and T. Tran. Heterogeneous web data search using relevance-based on the fly data integration. A. Mille, F. L. Gandon, J. Misselis, M. Rabinovich, and S. Staab, editors, *WWW*, pages 141–150, ACM, 2012.

[14] Y. Hu, K. Janowicz, G. McKenzie, K. Sengupta, and P. Hitzler. A linked-data-driven and semantically-enabled journal portal for scientometrics. In: *The Semantic Web - ISWC 2013*. Volume 8219 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013.

[15] D. Laniado and P. Mika. Making sense of twitter. P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks, and B. Glimm, editors, *International Semantic Web Conference (1).* Volume 6496 of Lecture Notes in Computer Science, pages 470–485, Springer, 2010.

[16] J. Letierce, A. Passant, J. Breslin, and S. Decker. Understanding how twitter is used to widely spread scientific messages. *Proceedings of the WebSci10: Extending the Frontiers of Society On-Line*, 2010.

[17] M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. SIGMOD ’10, pages 1155–1158, ACM, Indianapolis, Indiana, USA, 2010.

[18] G. Parra Chico and E. Duval. Filling the gaps to know More! about a researcher. *Proceedings of the 2nd International Workshop on Research 2.0. At the 5th European Conference on Technology Enhanced Learning: Sustaining TEL,* pages 18–22, CEUR-WS, September 2010.

[19] W. Reinhardt, M. Ebner, G. Beham, and C. Costa. How people are using twitter during conferences. *Hornung-Prähauser, V., Luckmann, M.(Hg.): 5th EduMedia conference, Salzburg*, pages 145–156, 2009.

[20] M. Schmachtenberg, C. Bizer, and H. Paulheim. Adoption of the linked data best practices in different topical domains. In: *The Semantic Web–ISWC 2014*, pages 245–260. Springer, 2014.

[21] S. Softic, M. Ebner, H. Mühlburger, T. Altmann, and B. Taraghi. Twitter mining #microblogs using #semantic technologies. *6th Workshop on Semantic Web Applications and Perspectives*, pages 1–9, 2010.

[22] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pages 990–998, 2008.

[23] K. Tao, F. Abel, Q. Gao, and G.-J. Houben. Tums: Twitter-based user modeling service. R. Garcia-Castro, D. Fensel, and G. Antoniou, editors, *ESWC Workshops*. Volume 7117 of Lecture Notes in Computer Science, pages 269–283, Springer, 2011.

[24] P. Thonhauser, S. Softic, and M. Ebner. Thought bubbles: a conceptual prototype for a twitter based recommender system for research 2.0. *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies.* i-KNOW ’12, pages 32:1–32:4, ACM, Graz, Austria, 2012.

[25] T. D. Ullmann, F. Wild, P. Scott, E. Duval, B. Vandeputte, G. A. Parra Chico, W. Reinhardt, N. Heinze, P. Kraker, A. Fessl, S. Lindstaedt, T. Nagel, and D. Gillet. Components of a research 2.0 infrastructure. *Lecture Notes in Computer Science,* pages 590–595, Springer, 2010.

[26] R. Van Noorden. Online collaboration: scientists and the social network. *Nature News*, 512(7513): 126–130, August 2014.

[27] K. Verbert, D. Parra, P. Brusilovsky, and E. Duval. Visualizing recommendations to support exploration, transparency and controllability. *Proceedings of the 2013 International Conference on Intelligent User Interfaces.* IUI ’13, 2013.

[28] K. Weller, E. Droge, and C. Puschmann. Citation analysis in twitter: approaches for defining and measuring information flows within tweets during scientific conferences. M. Rowe, M. Stankovic, A.-S. Dadzie, and M. Hardey, editors, *Making Sense of Microposts (#MSM2011)*, pages 1–12, 2011.

[29] R. S. Xin, O. Hassanzadeh, C. Fritz, S. Sohrabi, and R. J. Miller. Publishing bibliographic data on the semantic web using bibbase. *Semantic Web*, 4(1):15–22, 2013.

Chapter 8

Research Exploration Tool

In much of society, research means to investigate something you do not know or do not understand.

—Neil Armstrong.

This chapter describes the implementation of an approach for searching resources in the Web of Data for scientific research to demonstrate the use case described in Chapter 7. The implementation consists of two main components: a semantic search engine and an aligner. The semantic search engine takes care of the indexing and interpreting incoming queries, given a query context it ranks the found resources and presents them to the search interface. The aligner extracts, annotates, and interlinks the selected data sources.

8.1 Overview

Our approach is one of the first practical solutions combining the social web and the semantic web in an interactive search environment that visually emphasizes and represents the search context and results. We introduced the first data architectures in 2011 [8]. The data modeling concepts were discussed in [22, 24] while back-end components [6], were investigated. These components were used to serve the front-end implementation. The aligning and matching of research related semantic resources was the main scope of our work on dynamic alignment of scientific resources such as web collaboration tools and digital archives [9, 23]. The first prototypes of the search interface were introduced at conferences in late 2013 [7] and

2014 [23]. One of the first live versions was selected to participate at the Semantic Web Challenge 2013 at the International Semantic Web Conference [10]. The goal was to iteratively develop the use case implementation, demonstrate the interface and visualization, trigger discussion, and gain insight on the exploration workflow.

Figure 8.1 shows the different components:

(i) The Semantic Search Engine resolves queries consisting of one or more research concepts by being able to resolve them with refined entities out of Linked Data sets, represented in the model as "Data Seeds". The Semantic Search Engine parses queries and discovers relations between the research objects which are in fact a refined representation of the resources.

(ii) The Aligner allows configuring a selection and interlinking of structured data, linked data (semantically described structured data) and data from Social Media. The Aligner combines data from various heterogeneous sources configured in the Data Seeds and refines them for the Semantic Search Engine.

(iii) A search interface allows researchers to browse and search for new research objects based on the researcher's previous tracked research objects and traversed paths (such as bookmarks or saved searches).

Listing all relevant contributions of researchers improves the ranking of found resources related to a certain search. Combining the Aligner and the Semantic Search Engine is an essential aspect for this infrastructure. The semantic aspect (not shown in the figure) is essential for each search because it takes into account the meaning of the links between the resources. This meaning is documented in several (commonly used) vocabularies.

8.2 Data Seeds

There are three important data seeds:

(i) datasets derived from structured data;

(ii) linked data; and

(iii) data from social media.

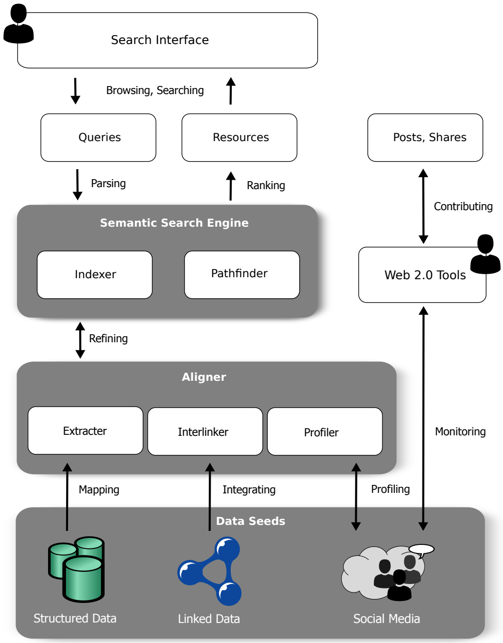


Figure 8.1: The combination of the Aligner and the Semantic Search Engine forms a bridge between the source data and researchers.

All this data is annotated with vocabularies. This section firstly introduces the used vocabularies and then explains the details about the datasets used.

Vocabularies

The Dublin Core vocabulary[1][29] has been used besides the Semantic Web for Research Communities (SWRC), the Semantically Interlinked Online Communities (SIOC) and the Friend-of-a-Friend (FOAF) ontology to annotate information such as titles, descriptions, authors, and other metadata properties.

Using the Modular Unified Tagging Ontology (MUTO)[2] [16] tags are annotated. MUTO is suitable because it combines and further optimizes succesful approaches from earlier tag ontologies. MUTO instances bind within the Linked Data hashtags and tags from Twitter and Mendeley within the same context. Instances of MUTO

[1] http://dublincore.org/documents/dcmi-terms
[2] http://muto.socialtagging.org/core

support interlinking tags with conference labels in Conference Linked Data[3] [22] (COLINDA).

Common vocabularies to annotate social media as Linked Data are: Friend of A Friend (FOAF)[4], Semantically Interlinked Online Communities (SIOC)[5] [3, 4], and Dublin Core[6] [29]. FOAF describes the user profiles, their social relations and resources. SIOC is mostly combined with FOAF and Dublin Core for creating instances of web entries like blogs, microblogs, mailing list entries, forum posts, along with other entries from Web 2.0 platforms [8, 24, 27]. Passant et al. improved mapping social profiles with related content, such as via interlinking tags [18, 19, 20].

Datasets

The selected datasets consist of existing Linked Open Data sets: DBpedia, DBLP and GeoNames interlinked with research oriented datasets such as COLINDA and a Social Linked Data set containing information about conferences and social profiles of the researchers from Twitter and Mendeley and the data they shared recently.

The "Digital Bibliography and Library Project" (DBLP)[7][15] provides bibliographic information on major computer science journals and proceedings and indexes more than 2.3 million articles. Besides it also has many links to home pages of computer scientists. The COLINDA data set resolves this connection. COLINDA describes conferences using the Semantic Web for Research Communities (SWRC)[8] ontology [26]. Especially important for this decision was that DBLP Linked Data also applies this ontology to describe its resources. COLINDA bridged GeoNames, DBpedia, and DBLP since it has links to these three Linked Datasets. Furthermore, it serves as a conference entity resolver for social data used with the profiles of users from Twitter and Mendeley.

[3] http://www.colinda.org

[4] http://xmlns.com/foaf/spec/

[5] http://rdfs.org/sioc/spec/

[6] http://dublincore.org/documents/dcmi-terms/

[7] http://dblp.l3s.de

[8] http://ontoware.org/swrc/

8.3 Semantic Search Engine

This module parses queries against the aligned data sources and ranks matched resulting resources. It consists of two modules: the *Indexer* and the *Pathfinder*. The Pathfinder retrieves resources via the Indexer. The Indexer pre-optimizes and stores each resource by uri and label to be able to serve them instantly. We have used an implementation that relies on our earlier work on pathfinding in linked data [6]. For all data sources we make sure that we describe their resources using correctly mapped and applied vocabularies so we can expose them using a uniform interface and representation, such as RDF.

8.4 Aligner

The Aligner module combines different social and online tools, such as Twitter or Mendeley. It interlinks data provided by the users (when they are actively using these social and personal media tools) to existing (Linked) Open Data such as DBpedia, GeoNames[9], LinkedGeoData[10][25], DBLP, and COLINDA. This interlinking allows enriching and connecting researchers to a vast amount of resources implicitly connected to them and thus initially not accessible. This allows to track communication on Social Media such as Twitter among researchers and relate it to publications and conferences. The Aligner module is optimized for the specificities of Social Media and collaboration tools. Moreover, a part of the alignment analysis, where access to restricted resources from users on Twitter and Mendeley is needed, happens on client-side. Only the results are aligned with the existing Linked Open Data.

Extracter

Each time when a certain source provides access to their structured content, the Aligner makes sure that provided content is correctly converted conform our data model. Therefore it selects configured properties and annotates them using the supported vocabularies.

[9] `http://www.geonames.org`
[10] `http://linkedgeodata.org`

Profiler

When users sign up, they authorize access to their Twitter and Mendeley accounts. The Profiler extracts the timeline and followers of the user's social account and then annotates them using the FOAF and SIOC vocabularies. Their author's profile is linked to DBLP based on publication title and the Digital Object Identifier (DOI) of each publication. Listing 8.1 shows how to combine these identifiers with all author names and use them to find matching author identifiers in DBLP for each publication. For each article in a Mendeley account linked to a subscribing researcher it checks the DOI and publication title in DBLP and retrieves the authors. If a match occurs, the articles are aligned using *owl:sameAs*. If all author names of the publication match, we interlink the Mendeley authors with the DBLP authors based on their URI's. Because users linked their Twitter and Mendeley when signing up, the profiler can link the author representation on DBLP with the author profile on Mendeley to the other social media accounts of the user and their contributions.

```
alignArticle(mendeleyArticle)
  title = find(mendeleyArticle, "dcterms:title")
  articleAuthors = aligner.getAuthors(title, article)
  foreach(articleAuthors -> (dblpArticle, authors))
    add(mendeleyArticle, "owl:sameAs", dblpArticle)
    foreach(authors -> (authorUri, authorName))
      add(articleUri, "dcterms:creator", authorUri)
      persons = find("foaf:name", authorName)
      foreach(persons -> person)
        add(person, "rdf:type", "foaf:Person")
        add(person, "owl:sameAs", uri)
```

Listing 8.1: Aligning research publications from Mendeley (mendeleyArticle) and DBLP (dblpArticle).

Including links to the social profiles of each researcher allows personalized searches. The resulting user profile extends the search context given a set of keywords.

Interlinker

Interlinking linked data involves several steps to optimally align various sources. The first step is to define the linked datasets to use, to identify the vocabularies in them and to define which resource to link with resources occurring in another dataset. If the dataset is not available as Linked Data, then we must select a vocabulary to annotate the data. The case of Social Media is particular because Social Media content often consists of small posts and shares which we analyzed based on:

- URLs referring to and the content in it (enriched with recognized entities);
- hashtags and mentions included;
- entities occurring with the tweets.

After we extracted the urls, hashtags, entities and mentions out of each post in Social Media, we checked each of those against the Linked Open Data Cloud. COLINDA is used for matching conference hashtags, LinkedGeoData and GeoNames for locations, DBpedia for general concepts such as persons, places, and events. DBpedia is the de-facto main hub within the LOD Cloud [2]. It is well-connected to the GeoNames and DBLP which makes it a very valuable source for search space expansion with more information about some common categories like cities and countries, persons or institutions. Additionally within the experiment DBpedia was also used as hub for path finding. We show an example for the hashtags in Listing 8.2: after loading the interlink services ("colinda","geonames","dbpedia","dblp") from a configfile in a list *interlinkServices*, we annotate each unique tag occurring in a microblogpost.

```
annotateTag(tag)
  labels = store.find(tag, "rdfs:label");
  foreach(labels -> (label))
    foreach(interlinkServices -> (service))
      meanings.add(getMeaning(service, label))
  store.add(tag, "muto:tagLabel", literal(label))
  store.add(tag, "muto:tagMeans", meanings)
```

Listing 8.2: Interlinking tags with the MUTO vocabulary. The tagLabel and tagMeans properties are used to indicate the label and a URI to the definition respectively.

Combining these approaches enriches tweets with Linked Data and is a good way to achieve optimal meaning. Entities occurring in the resources shared via the tweets lead to the best results [1]. However, we have found in earlier research work that also the hashtags have consistent enough meaning for interlinking [14].

Example of Interlinking Conferences

The *rdfs:seeAlso* property connects conferences from COLINDA with corresponding proceedings instances from DBLP Linked Data set. The *rdfs:label*, of each conference instance, matches the tags and hashtags from Social Media content and profiles of users. COLINDA instances also include the *dcterms:spatial*[11] property for venues

[11] http://dublincore.org/documents/dcmi-terms/#terms-spatial

of conferences found in DBpedia. Conference web page links are generated with the *owl:sameAs* property. The connection of the COLINDA spatial information to GeoNames [28], uses the *swrc:location* property. The description of the conference venue combines the GeoNames [12] ontology and basic Geo (WGS84 lat/long) Vocabulary [13] within the interlinking process. Data contained in COLINDA originates from WikiCfP [14] and Eventseer[15] and contains information about approximately 15000 conferences in the period from the year 2003 up to 2013.

Listing 8.3 shows, a sample instance of the *WWW 2012 conference*[16] in COLINDA. In order to interlink the DBLP instance of proceedings for each single conference

```
@prefix swrc: <http://swrc.ontoware.org/ontology#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://colinda.org/resource/conference/WWW/2012>
  a swrc:Conference ;
  rdfs:label "WWW2012" ;
  swrc:location <http://sws.geonames.org/2996944/> ;
  swrc:eventTitle "WWW 2012" ;
  rdfs:seeAlso <http://dblp.l3s.de/d2r/resource/publications/conf/www/2012> ;
  owl:sameAs <http://eventseer.net/e/16970/> ;
  dc:spatial <http://dbpedia.org/resource/Lyon>,
             <http://dbpedia.org/resource/France> ;
  swrc:startDate "2012-10-20"^^xsd:date ;
  swrc:description "21st international world wide web conference (WWW 2012)" .
```

Listing 8.3: Sample instance in COLINDA: WWW 2012 conference. The link to DBLP using the seeAlso and to DBpedia via the spatial property

we used URL structure features of both datasets. DBLP instances follow the URL pattern *conf/abbrevation/YYYY* to identify the corresponding conference e.g. **conf/www/2012** . This pattern in DBLP is also stored in the instance of *swrc:Proceedings* as separate *dc:identifier* property. The COLINDA URL pattern *conference/abbrevation/YYYY*, e.g., **conference/WWW/2012** identifies the same conference. Interlinking between the COLINDA and DBLP data sets is done by matching these two patterns. Further, names (labels) of locations of conferences in COLINDA were used in CURL requests and SPARQL queries against DBpedia and Geonames to interlink these values from COLINDA over *dcterms:spatial, swrc:location* properties with the corresponding elements in DBpedia and Geonames instances. Included conference

[12] http://www.geonames.org/ontology/

[13] http://www.w3.org/2003/01/geo/

[14] http://www.wikicfp.com

[15] http://eventseer.net

[16] http://colinda.org/resource/conference/WWW/2012

web page links were embedded into COLINDA instances using the *owl:sameAs* property.

The resolution of search results is based upon the properties of Linked Data instances like *rdf:label*, *owl:sameAs*, *rdf:seeAlso*, *dc:title*, *dc:spatial*, or *dc:description*. Those properties have been used in generation of Linked Data instances to preserve conference shortcuts (e.g., WWW2012), point to link of proceedings of a conference or, to connect alternative link about it, as well to literally describe the venues of scientific events.

8.5 Exploratory Interaction

Based on the ability of humans to rapidly scan, recognize, recall images, and detect changes in size, color, and shape, we aim to enhance the guidance of users during their search by using several visual aids of which the three most visible are:

1. **Shape:** We group sets of types in large groups and represent them using an embedded shape (an icon) or an outer shape. Figure 8.2 shows different icons assigned to a conference, location, and tag. Types that cannot be assigned a group are grouped in a category 'Miscellaneous'. The shapes help the user to distinguish between the types of offered results.

2. **Color:** Every entity has a type and associated unique color. For a certain result set the user gets an immediate impression of the nature of the found resources. Figure 8.3 depicts two different objects related to other objects and therefore have a different shape and size. On the left of the search interface there is a legend explaining the researcher the meaning of shapes and colors.

3. **Size:** Each entity is ranked according to novelty and relation to the context and sized according to the degree of attention they should attract. This is shown in Figure 8.4. The novelty quantifies the degree of being new, original or unusual. Particularly in this context it entitles resources that are remarkable and differ from the others because of their direct relations with neighbours or their semantics (in terms of occurring predicates). A goal of the search is to explore information not seen before which makes it difficult to define an accurate search goal. Besides allowing to search specific entities, the visualization facilitates exploratory browsing. This is particularly useful when searching for information with unclear defined search targets [17].

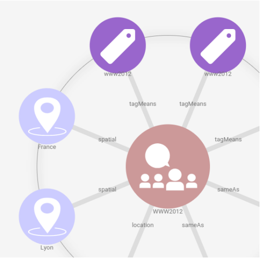


Figure 8.2: Different embedded shapes (icons) to distinguish types.

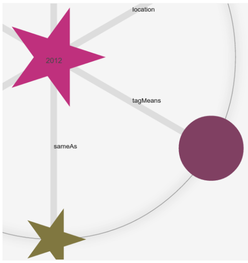


Figure 8.3: Different shape and color to distinguish types.

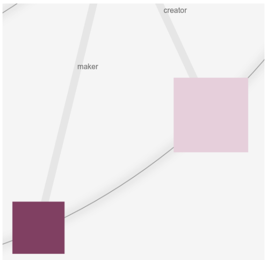
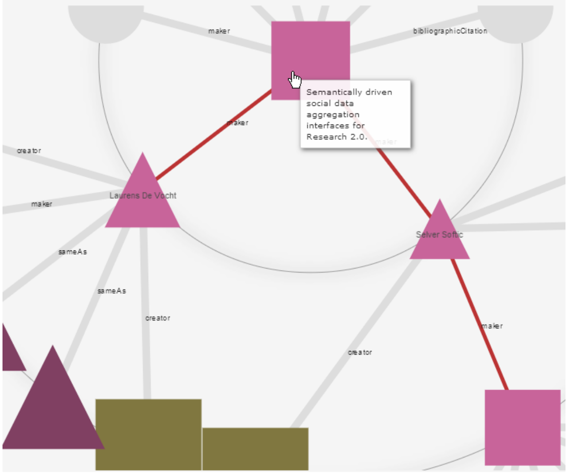


Figure 8.4: Different sizes to guide the user's focus.

Figure 8.5 shows how researchers can track the history of their search: the explored relations are marked red and clearly highlight the context of a search. This is a good example of how our system adapts to the users and their environments. It shows one of the ways how to build a model of the goals and knowledge of an individual user [5], and the model is used throughout the interaction with the user. Researchers can click on a list of resources they have searched to focus the visualization. A screencast of the search interface is available online[17]. In this screencast, we show how researchers interact with the search interface and the above described visualization.



Semantically driven social data aggregation interfaces for Research 2.0.

Figure 8.5: An emphasized line marks the explored relations in the visualized search context.

[17] http://youtu.be/tZU97BQxE-0

Example Illustrating the Dynamics

Each search starts within the search interface where a user can either login or query anonymously the Semantic Search Engine. The search interface distinguishes between two types of queries: a query which consists of several keywords as seeds and a profile-driven query, used as preset for further search, driven initially by user background information. We have developed a prototype, called ResXplorer to demonstrate the search engine [7]. ResXplorer presents search results according to the principles of a topological radial graph interface [30].

Except for the first step, the querying paradigm applies to the personalized search as well. The query in the figure illustrates the common case where a researcher enters the search process by entering simple keywords and tries to resolve the context of “finding useful resources from a certain conference”.

1. One searches for a specific conference “Linked Data" and articles related to “WWW2012". Firstly, as Figure 8.6 shows, the visualization focuses first on the logged in user upon which the user can choose to expand on of the neighbouring resources.

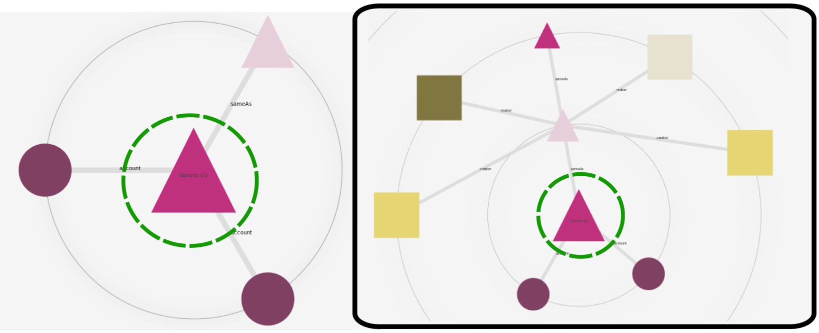
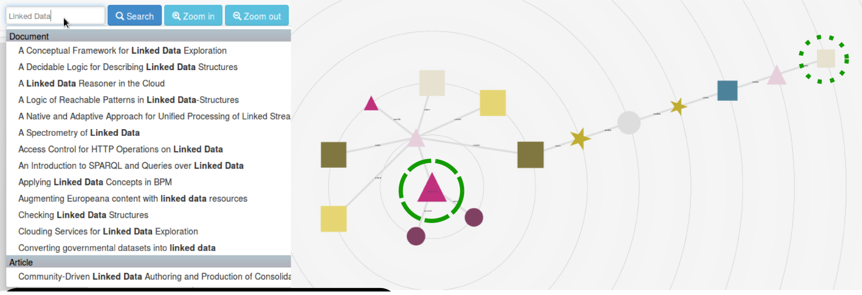


Figure 8.6: User expands a direct neighbor after ResXplorer focuses on logged in user (encircled with dashes).

   We note that the user changes the focus of the result view by clicking on a resource: the resource encircled with a mixed line. Within the simplified query progression process, entered keywords are first mapped towards the entities and properties in the index.

2. As a search result the engine delivers first set of links for each keyword entered, such as in Figure 8.7 for “Linked Data".



Document

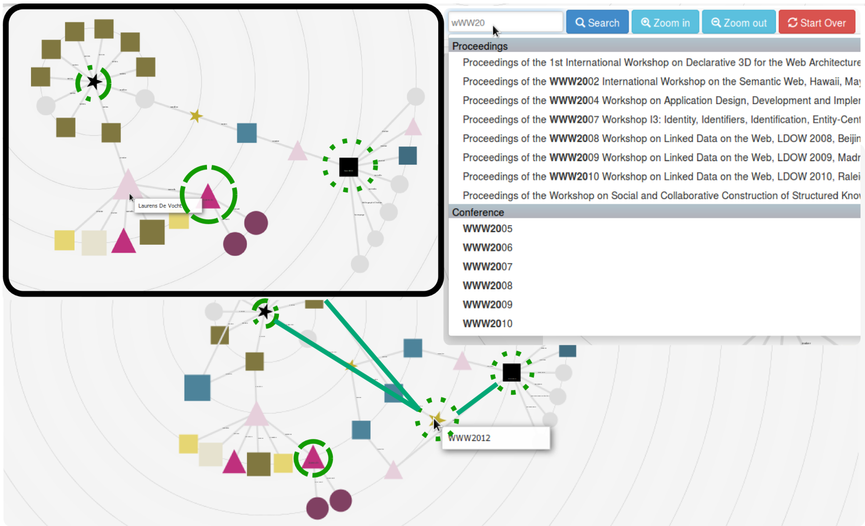
A Conceptual Framework for **Linked Data** Exploration
A Decidable Logic for Describing **Linked Data** Structures
A **Linked Data** Reasoner in the Cloud
A Logic of Reachable Patterns in **Linked Data**-Structures
A Native and Adaptive Approach for Unified Processing of Linked Strea
A Spectrometry of **Linked Data**
Access Control for HTTP Operations on **Linked Data**
An Introduction to SPARQL and Queries over **Linked Data**
Applying **Linked Data** Concepts in BPM
Augmenting Europeana content with **linked data** resources
Checking **Linked Data** Structures
Clouding Services for **Linked Data** Exploration
Converting governmental datasets into **linked data**

Article

Community-Driven **Linked Data** Authoring and Production of Consolida

Figure 8.7: User searches for "Linked Data" and ResXplorer reveals the chain of links between the selected document (encircled with dots) and the user.

3. If available, the system also delivers the types of entities discovered in index. When the user searches for the next keyword "WWW2012", relations to other already visualized resources are exposed as indicated in Figure 8.8.



Proceedings

Proceedings of the 1st International Workshop on Declarative 3D for the Web Architecture
Proceedings of the **WWW20**02 International Workshop on the Semantic Web, Hawaii, May
Proceedings of the **WWW20**04 Workshop on Application Design, Development and Impler
Proceedings of the **WWW20**07 Workshop I3: Identity, Identifiers, Identification, Entity-Cent
Proceedings of the **WWW20**08 Workshop on Linked Data on the Web, LDOW 2008, Beijin
Proceedings of the **WWW20**09 Workshop on Linked Data on the Web, LDOW 2009, Madr
Proceedings of the **WWW20**10 Workshop on Linked Data on the Web, LDOW 2010, Ralei
Proceedings of the Workshop on Social and Collaborative Construction of Structured Know

Conference

WWW2005
WWW2006
WWW2007
WWW2008
WWW2009
WWW2010

Figure 8.8: After the user focused on a common resource of both and searched for "WWW2012", ResXplorer reveals relations to the selected conference "WWW2012" (encircled with dot-dashes).

4. By entering the location, for example "Germany", one could narrow further the focus of the context by location. Each time a combination of various resources is visualized, the application suggests new queries: they are generally most useful for refining the system's representation of the researcher's need.

In case they have no idea which entity to focus on or what topic to investigate next they get an overview of possible entities of interest, like points of interest on a street map. By profiling their activities and contributions on social media and other platforms such as their own research publications, the affinity with the proposed resources is enhanced.

5. With each further iteration the user can choose either one of two actions:

   - **Query Expansion:** The user expands the query space by clicking the results retrieved by initial keyword based search.

   - **Additional Query Formulation:** Additional query expansion happens either through adding further keywords as well as through keyword combinations already entered where the back-end tries to deliver additional results based upon connection paths between the resources. What happens in return is that the engine tries to identify the terms that have been searched in the result space. In cases when they can be resolved by a Linked Data instance, the algorithm continues step by step looking via links to the neighbors of the instance to find a path to other terms identified by the engine as well. After a certain number of steps (here, seven) it terminates if it is unsuccessful.

8.6 Evaluation

We compared the implementation against both popular academic search engines and highly specialized academic search interfaces and evaluated the visualization itself by measuring the efficiency and average precision of the results presented to users.

We used a task based approach as already applied [12] to obtain expert user reviews. The goal of the reviews is to compare ResXplorer against industry reference academic search interfaces and related academic projects, the state-of-the-art (SOTA). Two researchers – search interface experts – independently reviewed the performance of each of these search interfaces. They were familiar with all of the tools beforehand. We selected a set of six representative tasks supported by these systems for the reviews in Table 8.1.

Table 8.1: List of tasks executed by the expert users.

Task	Description
T_1	Find proof that Chris(tian) Bizer is an author.
T_2	Find out three different people that know or are known by the person in T1 (e.g. co-authors).
T_3	Find out three different kinds of relations between the person in T1 and Chris(tian) Bizer.
T_4	Find three different conferences on the subject Artificial Intelligence.
T_5	Find at least two people that have a paper included in the proceedings in two consequent editions of the WWW (World Wide Web) Conference.
T_6	Find: (i) at least one publication that was presented in 2011 in a WWW workshop (co-)organized by Tim Berners-Lee (e.g. LDOW - Linked Data on the Web); and (ii) at least one publication with an author that relates this publication to both the ‘2011 publication and the ISWC Conference 2010.

We designed the search tasks optimized for the SOTA search engines and for ResXplorer and they are either simple (e.g. single fact or source) or complex (combinations of facts and sources). We outlined the a priori, thus before presenting it to the expert users, expected suitability of these tasks in Table 8.2.

Table 8.2: *A priori* optimal suitability of the search tasks.

	Straightforward	Complex
ResXplorer	T_3	T_6
Both	T_2, T_4	
SOTA	T_1	T_5

In each of these tasks the experts had to indicate after each interaction by either a click or text input, how many relevant results they found. Their actions were recorded so that we could count the total number of actions for each task and the number of results after each action.

For each of the tasks we measured the *average precision* (between 0 and 1) and the *efficiency* (expressed as number of actions needed).

Average Precision, measures the average of the search precision over all the required actions in certain task. Thereby the precision [21] of the kth search action is defined for the user evaluation as follows:

$$\textbf{precision} = P_k = \frac{\text{retrieved relevant results}}{\text{retrieved results}} @k \tag{8.1}$$

and the average precision over all actions A in certain task:

$$\textbf{average precision} = AP = \sum_{k \in A} \frac{P_k}{|A|} \tag{8.2}$$

However, the actions are different so a direct comparison for ResXplorer between the user action effectiveness and the precision measured here is not possible. It also would make no sense as the user tests focused on lean users while the experts are specialized in search interfaces.

Efficiency, expressed as the number of actions (N_x) when users perform a certain task (T_x). The lower the score, the less actions the experts needed to successfully complete the task.

To verify that the expert reviews are similar enough to be considered, we measured the inter-rater agreement among them. We selected therefore the *chance corrected agreement* (κ) measure [11] ($-1 < \kappa < 1$). The inter-rater agreement of the results between the experts is substantial ($\kappa = 0.61$ and F-measure 0.83) according to the Landis et al. scale [13]. The visualization in Figure 8.9 shows the mean average results for each of the tested search interfaces and indicates how well the expert reviews match.

Tables 8.3 and 8.4 display the results of the expert evaluations of ResXplorer in comparison to two industry references and three research projects in the same domain. In ARNet Miner and ResXplorer the autocomplete facilitated instant and precise matches. In Microsoft Academic Search, Google Scholar, and Falcons the first page of results contained the necessary results and Google Scholar and Microsoft Academic Search promoted the matching result as a suggestion on top of the list.

T3 is a non-direct relation finding task and that is the main goal of ResXplorer while T2 requires zooming in depth around a specific property of a person. ResXplorer

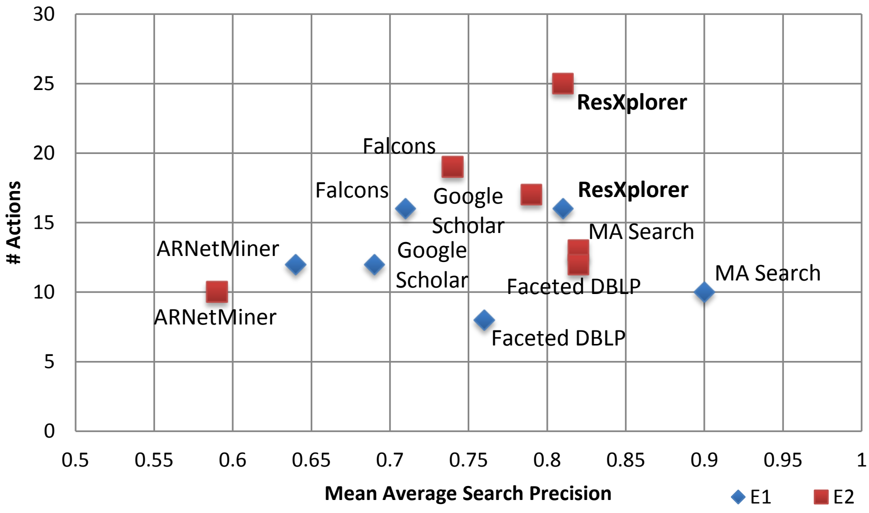


Figure 8.9: The agreement between the experts on the ratings over all search interfaces combined is substantial. (*E1 = expert 1, E2 = expert 2*)

Table 8.3: The search precision for getting the first search results returns all true positive matches except ArnetMiner returned 4 out of 5 false positives in T_1. ResXplorer is not as precise as the other interfaces for T_2 but excels in T_3. *(brighter = better)*

Effectiveness	T_1	T_2	T_3	T_4	T_5	T_6	Mean
Google Scholar	1.00	0.90	0.35	1.00	0.43	0.62	0.72
MA Search	1.00	1.00	0.63	1.00	0.90	0.64	0.86
Falcons	1.00	0.95	0.63	0.78	0.60	0.68	0.77
ResXplorer	1.00	0.84	0.84	0.70	0.39	0.80	0.76
ARNetMiner	0.60	1.00	0.81	0.74	0.20	0.49	0.64
Faceted DBLP	1.00	1.00	0.83	0.95	0.52	0.45	0.79

Table 8.4: An increased number of user actions does not always guarantee more precise (intermediate) results, but it does for ResXplorer, except in T_5. *(brighter = better)*

Efficiency	T_1	T_2	T_3	T_4	T_5	T_6	Sum
Google Scholar	1	1	2	2	5	3	15
MA Search	1	1	3	1	2	4	12
Falcons	1	2	3	3	3	6	18
ResXplorer	1	2	4	3	4	4	21
ARNetMiner	1	1	3	1	2	3	11
Faceted DBLP	1	1	3	1	2	3	10

intends to maintain the broad overview at all times during the search which induces some noise for a task like T2.

In T4 the industry references beat the research engines. T4 requires skimming or filtering a list of conferences which is not supported in ResXplorer and in Falcons and ArnetMiner not to the same degree as the industry references. Faceted DBLP also scores well for T4 thanks to the faceted search interface and tight DBLP link. For T4 required the Google Scholar interface scrolling through two pages to find three different conferences. There were many results of the same conference on a page. Microsoft Academic Research allowed searching specifically for items of the type conference. That explains the highest rating here, as all results were on the first page in contrast to Google Scholar. In Falcons the results were a little less accurate and did not allow searching specifically for conferences either. ResXplorer did not provide a list but a limited set of entry points for exploration. This meant the search was repeated to find different entry points leading to a conference, in fact three times, each time to find a new conference. ARNet Miner provided a view of the results containing distracting widgets, not all material was clearly relevant for the search. It included relatively many false positives to interpret but all results were found after one search action. The expert users judge the results presented in the *a priori* defined complex tasks having the most irrelevant results and they needed at least 2 actions in T5 and even 3 actions in T6 to resolve the search task. The highest effectiveness was found for MA Search in T5 and for ResXplorer in T6. In terms of efficiency Google Scholar required the most actions in T5 and Falcons in T6.

8.7 Discussion

The evaluation presented a balanced choice of comparable solutions for the same or closely related use cases: two of them from industry (MA Search and Google Scholar) and three of them from research domain (ARNet Miner, Falcons and Faceted DBLP). This allowed good positioning and qualitative reviewing of our use case implementations. ResXplorer is situated in the mid-range in terms of mean average search precision and requires relatively lots of action from the user. However, ResXplorer is best when the task consisted of relating resources that are not directly related or when at least the user is not aware of how they are related. That is precisely the goal we wanted to show with ResXplorer and the methods and techniques that drive it.

Room for Improvements

The main concept of ResXplorer resides on the idea of an interactive search interface which leads the researcher through the process of expansion and exploration of results to the hidden implicit valuable information discoveries which are uncovered in such a process. To make ResXplorer more precise in classical search and retrieve scenarios, more accurate filters on the search keywords and results are crucial. Analyzing nuances concerning the efficiency would be beneficial as a smaller number of actions does not always lead to the most efficient interface, certainly if it requires more thinking and judging from the users: more straightforward steps might be more efficient than less but more complicated steps. The distinction between proposing new affinities between certain resources versus exploring the proposed resources in detail could be more clear and explaining the motivation behind the affinities, where we characterize each affinity, between researchers and resources, by the amount of shared interests and other commonalities.

Contributions

With this implementation users can combine any searches and interact with the results that exposes relationships between them. This is a feature not found in conventional search interfaces. It offers search for publications, as well as supports relation visualization on author level. We visually emphasize discovered types of entities and relations. In comparison to the current existing solutions we can use the snapshot of social content published by researchers on social media and collaborative platforms like Twitter and Mendeley to make a pre-set for exploratory search. This feature is unique to our solution. Furthermore, the method by which we generate context-based results differs from ARnet Miner because we do not rely on data mining and machine learning techniques to resolve the research related information. Our approach uses affinity based ranking derived from the social context and search process itself. We use graph based algorithms which perform independently of underlying Linked Data. In comparison to the existing search solutions, our interface is designed to visually explore the research space, rather than to support classical keyword based search. This exploration is based on personal preference and serendipity of information in the data set (publications, persons, events). This data is enhanced by additional information (e.g. venues of events) related to the search. Unlike Microsoft Academic Search and ARnet Miner our graph visualization is expandable and includes entities from Linked Data and description of relations between them. Since pre-sets of the search reside on actualized social media content of the user our solution adapts better on changes of information and trends from social media. This aspect differs strongly from the conventional approaches mentioned here.

Summary

The resulting semantic search application provided both a technical demonstration as well as an interactive visualization of search results. The main contribution is, besides retrieving resources from selected Linked Data repositories, allowing researchers to interactively explore relationships between the resources and entities like events or persons related to their work. In particular, when a part of the search consists of finding resources that connect a given statement, such as finding common items between two authors of an article, the implementation delivered the relatively highest search precision. Further improvements on the ranking criteria should improve the precision of proposed affinities and the results even further.

References

[1] F. Abel, Q. Gao, G.-J. Houben, and K. Tao. Semantic enrichment of twitter posts for user profile construction on the social web. In: G. Antoniou, M. Grobelnik, E. Simperl, B. Parsia, D. Plexousakis, P. Leenheer, and J. Pan, editors, *The Semanic Web: Research and Applications*. Volume 6644 of Lecture Notes in Computer Science, pages 375–389. Springer Berlin Heidelberg, 2011.

[2] S. Auer, J. Demter, M. Martin, and J. Lehmann. Lodstats - an extensible framework for high-performance dataset analytics. A. ten Teije, J. Volker, S. Handschuh, H. Stuckenschmidt, M. d'Aquin, A. Nikolov, N. Aussenac-Gilles, and N. Hernandez, editors, *EKAW*. Volume 7603 of Lecture Notes in Computer Science, pages 353–362, Springer, 2012.

[3] J. G. Breslin, S. Decker, A. Harth, and U. Bojars. Sioc: an approach to connect web-based communities. *International Journal of Web Based Communities (IJWBC)*, 2(2):133–142, 2006.

[4] J. G. Breslin, A. Harth, U. Bojars, and S. Decker. Towards semantically-interlinked online communities. A. Gomez-Perez and J. Euzenat, editors, *European Semantic Web Conference (ESWC)*. Volume 3532 of Lecture Notes on Computer Science, pages 500–514, Springer, 2005.

[5] P. Brusilovsky and R. Rizzo. Using maps and landmarks for navigation between closed and open corpus hyperspace in web-based education. *New Rev. Hypermedia Multimedia*, 8(1):59–82, January 2003.

[6] L. De Vocht, S. Coppens, R. Verborgh, M. Vander Sande, E. Mannens, and R. Van de Walle. Discovering meaningful connections between resources in the web of data. *Proceedings of the 6th Workshop on Linked Data on the Web (LDOW2013)*, Rio de Janeiro, Brazil, 2013.

[7] L. De Vocht, E. Mannens, R. Van de Walle, S. Softic, and M. Ebner. A search interface for researchers to explore affinities in a Linked Data knowledge base. *Proceedings of the 12th International Semantic Web Conference Posters & Demonstrations Track*. CEUR-WS, pages 21–24, 2013.

[8] L. De Vocht, S. Softic, M. Ebner, and H. Mühlburger. Semantically driven social data aggregation interfaces for research 2.0. *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*. i-KNOW 2011, pages 43:1–43:9, ACM, Graz, Austria, 2011.

[9] L. De Vocht, S. Softic, E. Mannens, M. Ebner, and R. Van de Walle. Aligning web collaboration tools with research data for scholars. *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*. WWW Companion 2014, pages 1203–1208, International World Wide Web Conferences Steering Committee, Seoul, Korea, 2014.

[10] L. De Vocht, S. Softic, E. Mannens, R. Van de Walle, and M. Ebner. Resxplorer: interactive search for relationships in research repositories. *International Semantic Web Conference : Semantic Web Challenge, Abstracts*, pages 8, Trentino, Italy, 2013.

[11] G. Hripcsak and A. S. Rothschild. Agreement, the F-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12(3):296–298, 2005.

[12] W. Kraaij and W. Post. Task based evaluation of exploratory search systems. *SIGIR 2006 workshop, Evaluating Exploratory Search Systems*, 2006.

[13] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 1977.

[14] D. Laniado and P. Mika. Making sense of twitter. P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks, and B. Glimm, editors, *International Semantic Web Conference (1).* Volume 6496 of Lecture Notes in Computer Science, pages 470–485, Springer, 2010.

[15] M. Ley. The DBLP computer science bibliography: evolution, research issues, perspectives. *String Processing and Information Retrieval.* Springer, pages 1–10, 2002.

[16] S. Lohmann, P. Diaz, and I. Aedo. Muto: the modular unified tagging ontology. C. Ghidini, A.-C. N. Ngomo, S. N. Lindstaedt, and T. Pellegrini, editors, *I-SEMANTICS.* ACM International Conference Proceeding Series, pages 95–104, ACM, 2011.

[17] S. Pace. A grounded theory of the flow experiences of web users. *International journal of human-computer studies*, 60(3):327–363, 2004.

[18] A. Passant, U. Bojars, J. G. Breslin, and S. Decker. The sioc project: semantically-interlinked online communities, from humans to machines. J. A. Padget, A. Artikis, W. Vasconcelos, K. Stathis, V. T. da Silva, E. T. Matson, and A. Polleres, editors, *Coordination, Organizations, Institutions, and Norms in Agent Systems V.* Volume 6069 of Lecture Notes in Computer Science, pages 179–194, Springer, 2009.

[19] A. Passant, U. Bojars, J. G. Breslin, T. Hastrup, M. Stankovic, and P. Laublet. An overview of smob 2: open, semantic and distributed microblogging. W. W. Cohen and S. Gosling, editors, *ICWSM*, pages 303–306, The AAAI Press, 2010.

[20] A. Passant, J. G. Breslin, and S. Decker. Open, distributed and semantic microblogging with smob. B. Benatallah, F. Casati, G. Kappel, and G. Rossi, editors, *ICWE.* Volume 6189 of Lecture Notes in Computer Science, pages 494–497, Springer, 2010.

[21] D. M. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[22] S. Softic, L. De Vocht, E. Mannens, M. Ebner, and R. Van de Walle. COLINDA: modeling, representing and using scientific events in the web of data. *Proceedings of the 4th International Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2015) Co-located with the 12th Extended Semantic Web Conference (ESWC 2015), Protoroz, Slovenia, May 31, 2015.* pages 12–23, 2015.

[23] S. Softic, L. De Vocht, E. Mannens, R. Van de Walle, and M. Ebner. Finding and exploring commonalities between researchers using the resxplorer. *Learning and Collaboration Technologies. Technology-Rich Environments for Learning and Collaboration*, pages 486–494, Springer International Publishing, 2014.

[24] S. Softic, M. Ebner, H. Mühlburger, T. Altmann, and B. Taraghi. Twitter mining #microblogs using #semantic technologies. *6th Workshop on Semantic Web Applications and Perspectives*, pages 1–9, 2010.

[25] C. Stadler, J. Lehmann, K. Höffner, and S. Auer. Linkedgeodata: a core for a web of spatial open data. *Semantic Web*, 3(4):333–354, 2012.

[26] Y. Sure, S. Bloehdorn, P. Haase, J. Hartmann, and D. Oberle. The SWRC ontology - semantic web for research communities. *Progress in Artificial Intelligence*, 2005.

[27] K. Tao, F. Abel, Q. Gao, and G.-J. Houben. Tums: Twitter-based user modeling service. R. Garcia-Castro, D. Fensel, and G. Antoniou, editors, *ESWC Workshops*. Volume 7117 of Lecture Notes in Computer Science, pages 269–283, Springer, 2011.

[28] R. Volz, L. Zhou, T. Finin, and A. Joshi. Towards ontology-based disambiguation of geographical identifiers. *I3: Identity, Identifiers, Identification Workshop, World Wide Web Conference (WWW 2007)*, pages 8–12, 2007.

[29] S. Weibel. The dublin core: a simple content description model for electronic resources. *Bulletin of the American Society for Information Science and Technology*, 24(1):9–11, 1997.

[30] K.-P. Yee, D. Fisher, R. Dhamija, and M. Hearst. Animated exploration of dynamic graphs with radial layout. *Proceedings of the IEEE Symposium on Information Visualization*. Volume 43 of 2001.

Part IV

Conclusions

Chapter 9

Conclusions

This thesis aims to facilitate exploring semantic relationships between resources on the Web of Data. To this end, I developed and evaluated three techniques, each focusing on a separate aspect of exploratory search: front-end, back-end, and the bridge in between. The techniques involve linking data and semantically modeling structured data to make data more discoverable via the Web. The fundamental concepts were outlined in Chapter 3. In this final chapter, I revisit the research questions and indicate possibilities for future research.

9.1 General Findings

The thesis focused on two cases of exploring data on the Web, when users:

(i) how to formulate a search query (e.g., which keywords to include) to find exactly what they are looking for;

(ii) want to browse information instead of looking up something specific.

The theoretical background, mainly by White and Marchionini et al., explained exploratory search as a shift from query-to-document matching to direct guidance at all stages of information-seeking [2]. They stated that exploratory search is likely to enable better understanding of a problem context, allowing searchers to make more informed decisions about interaction or information use [3] and emphasized the relation between the different aspects of user-system interaction during exploratory search: lookup, investigate and learn [1].

Many past projects focused on one or more of these aspects. Related work on this subject appeared to be focused either mainly on the semantic modeling, on interactivity of the exploration or were implementations for a specific application. Chapter 2 lists some important relevant work. The distinct support for search scenarios, where the user was not aware of the revealed relationships between resources, is one of the main contributions of this thesis. Like some other implementations, the proposed techniques in this thesis rely on controlled vocabularies and the semantic descriptions of connections between resources to drive the implementation of each technique. The main difference is that the proposed techniques in this thesis emphasize the exploration of relationships between resources rather than a more detailed exploration of a specific resource. Finally, I combined the techniques and applied them to a use case about aligning digital libraries, scientific conferences and researchers. The opportunities lie in applying the techniques to combinations of different linked data sources, covering an entire workflow ranging from back-end to front-end, without denormalizing the semantics along the way.

9.2 Answers to the Research Questions

Overall, supporting exploration on top of linked data should turn the potential of its exploitation more likely, while at the same time allowing users to discover the data. In Chapter 2, I outlined the research questions with their hypotheses and the key objectives of the study. This PhD answered the questions across three different chapters, each dealing with one of the proposed techniques: **Chapter 4** – Interactive Search Visualization; **Chapter 5** – Query Processing; **Chapter 6** – Path-based Storytelling.

Finally, more evidence was gathered by combining and applying the techniques to a use case, detailed in **Chapter 7**. The implementation and evaluation is described in **Chapter 8**. I investigated there how accurate a research exploration tool facilitates visually exploring linked data, without providing a 'traditional' ranked list of results.

The first and main research question **RQ1** arose when looking at the combination of exploratory search on the Web and the new *graph* structure (consisting of *triples*) that the Semantic Web brought. This allowed linking data to each other, more fine-grained than with hypertext documents or APIs. On top of that, the graph structure and the (indirect) relationships it brought forth seemed like an obvious addition to search.

> *Can exploratory search efficiently and adequately address the user's intent when revealing relationships between resources?*

The results in Chapter 8 indicate that ensuring that retrieved relationships are adequate has a trade-off in terms of efficiency. Optimizing both at the same time is not possible without giving in on other areas such as: effort required from the user (evaluated in Chapter 5), generic applicability or control over the kind of relationships (tested in Chapter 6). When it comes to efficiency, the performance evaluation in Chapter 5 showed a linear execution time (scaling with increasing number of hops between resources) and an optimized space complexity, due to the introduction of a heuristic. Chapter 6 explains the internals and the performance of the underlying pathfinding algorithm in combination with linked data, more specifically the use of the A* algorithm. Choosing a suitable heuristic ensures keeping the execution time to scale linear with the search space, or in other words the more indirect and the more hops in between two resources the larger the number of resources needed to be checked for each query became. The evaluation of the prototype of the research exploration tool in Chapter 8 shed more light on the adequacy of addressing the user's intent. It mainly excels in this area compared to related work, when search tasks require relations among two or more items. But it comes with a trade-off in terms of user interaction.

The findings about the trade-off are strongly related with how the user's actions influence the results, the subject of the next research question **RQ2**.

> *To what degree do users' search actions influence the relevance and precision of search results?*

I found that each type of user action has a completely different influence on the relevance and the precision of search results. Chapter 4 taught that the processing of queries and the mapping of keyword queries proved to be of promising precision, given the complex and dynamic nature of the used datasets: a combination of Linked Open Data and non-linked data sources. I observed that searching by keywords for resources increases the result set with more new relevant resources, while it is on average as precise as expanding existing resources in the result set. The most precise user action was adding top-related nodes to the result set. The evaluation of the prototype of the research exploration tool in Chapter 8 showed that when comparing it to existing industry references (Google Scholar, Microsoft Academic

Search) and more experimental research projects (e.g. ArnetMiner, Falcons, Faceted DBLP), the implementation requires more interaction from the user and performs in the mid-range for search and retrieve scenarios.

The decision and actions the user is able to take depends on the results presented to them. Research question **RQ3** looks into this, when a user searches for two or more resources the relationship between them may be exposed, but it is unclear how this facilitates the exploration.

> *How does a justification of the presented results influence the user's certainty in getting closer to achieving the search goal?*

Firstly, the answer on the 'how'-part of the question relies mainly on the assessment of the technique developed for revealing relations between Linked Data resources using path-based storytelling in Chapter 6. To obtain interesting paths the technique enables optimizing the graph weights between resources and to take into account the semantics of the linked properties (*predicates*). I tried different combinations of weights and heuristics to find out how they affect the search results. The test-results with the DBpedia dataset indicated that the link estimation for relationships by the proposed path-algorithm is dependent on the choice of heuristic and weights. Finally the answer on the 'getting closer to the goal'-part is tricky because aiming for a relatively lower number of user actions for exploratory search does not always lead to the most efficient search approach. The evaluation in Chapter 8 showed that advanced exploration features (in this case driven by the path-based storytelling technique) could come at the cost of dropping basic features (or making them hardly accessible). This is mainly due to the narrow focusing on the exploration aspect of search, mismatching the expectation of some users looking for a more traditional 'search and filter' approach. Certainly if interacting requires more thinking and judging (search results) from the users: in this case more familiar, straightforward steps might be more efficient than less but more complicated steps.

This leads to the last research question **RQ4** looking into how search actions actually influence the search results.

> *How do users gradually refine a search query by interacting with its search results?*

To be able to gradually refine a search query, Chapter 4 proposes a workflow where users start broad until their desired detail level to then additionally explore until they reached their search task's goal or are satisfied with their discoveries. The workflow was evaluated in chapters 4, 5, and 8 where it came out as at least a facilitator for exploratory search. Chapter 4 presented the workflow and introduced three parts to guide the user interaction with the search results. The narrowing part and coordinated view proved to be helpful in terms of productivity for them in discovering and exploring the linked data published in a dataset. The broadening part helps users to find new insights and further expand links, in particular exposing direct neighbours following a specific search query added the most relevant resources to a result set. Taking into account the semantics of relationships between the visualized search results, I concluded that this kind of visual workflow at least enables users to discover and explore the information. This is further backed by the evaluation of the search precision over consequent actions in Chapter 5, each individual link builds a novel connection of potential interest to users and this number of connections does not necessarily have to be high compared to the total number of resources. The proposed engine is more precise than a raw SPARQL query baseline for query well-defined contexts, when it consists of keywords with an unambiguous meaning to both users and machine. The research exploration tool, from the use case implementation in Chapter 8, is situated in the mid-range, but requires more interaction from its users, when compared to related industry and academic projects, with similar goals and the same target audience. However, the implementation delivered relatively the highest search precision when a part of the search consisted of retrieving relationships. These relationships facilitate figuring out how newly added resources are connected to one or more existing results, such as finding common items between two authors of an article.

9.3 Future Work

Because my approach remains close to the linked structure of the data, this method is applicable to other domains when adequately structured, for example by aligning the selected vocabularies to the used datasets accordingly. All proposed techniques contribute to the authenticity of the semantically modeled data (after preprocessing, indexing). This means that they process queries and results by guaranteeing that the final output towards the user has useful results in its domain of application. However, in this PhD thesis, the techniques were tested mainly with data about encyclopedic facts derived from Wikipedia (DBpedia), data from academic libraries (DBLP) and data from social media (Mendeley, Twitter). A crucial next step is to repeat some of the experiments with data from a different domains such as among others biomedical, heritage, tourism and travel data.

Furthermore, it is likely that the user at a certain point may desire to configure the nature of the semantic connections in the discovered relationships. This implies extending and modifying the currently investigated optimizations of the path-based storytelling technique. The optimization now stands in the way of guaranteeing that the found path is the most suitable path for a specific context as this is not being taken into account. For example: when a specific path would be of interest for children rather than adults in a museum; or in a biomedical context where relations want to be exposed and take into account the background of the expert who is querying. Such context sensitive paths would require modifying the link weights and heuristics each time given a new target audience such that preference can be given to a more suitable path, rather than a default (not user specific) configuration.

To obtain a more nuanced view on the impact of the different weights and heuristics I plan to repeat the experiments in Chapter 6 with different datasets from different domains and a larger amount of test queries. This will allow to focus on investigating the correlation between the effect of the link estimation on the arbitrariness as perceived by users and the used computational semantic relatedness measures. One way to do this is to present users series of pairs of concepts, that define the story context; then present the matching stories after using different weights and heuristics for each query; and finally ask users to rate each story (select their preference) and to indicate how arbitrary and how relevant the story is given the context.

Finally, another important aspect is including details that facilitate users to obtain a more sophisticated selection and linking of contributed resources based on previous assessments and explored links. The focus of the algorithm until now mainly lay in the broadening aspect while narrowing reused existing approaches or only focused on retrieving more details. To enable this, we will have to modify the path-based storytelling algorithm to take into account context parameters and (user) feedback from retrieved results during the search. This opens up the possibility to make the algorithm (self) learning. The goal would be to effectively limit the search space, instead of a heuristically optimized path where the heuristic is mainly topological and not taking into account any possible tweaks based on the already found results, that is the information if the results are actually relevant or not. This thesis focused on retrieving semantical coherent relationships and aimed for high serendipity, with a decent 'affinity' between the resources: not too trivial and not too arbitrary.

References

[1] G. Marchionini. Exploratory search: from finding to understanding. *Commun. ACM*, 49(4):41–46, April 2006.

[2] R. W. White, G. Muresan, and G. Marchionini. Evaluating exploratory search systems. *Proceedings of the ACM SIGIR 2006 Workshop on Evaluating Exploratory Search Systems*, pages 1–2, 2006.

[3] R. W. White and R. A. Roth. Exploratory search: beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1):1–98, 2009.