Linked data based exploratory search

Nicolas Marie

UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS

ÉCOLE DOCTORALE STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

T H È S E

pour obtenir le titre de

Docteur en Sciences

de l'Université de Nice - Sophia Antipolis
Mention : INFORMATIQUE

Présentée et soutenue par

Nicolas MARIE

Linked data based exploratory search

Thèse dirigée par Fabien GANDON préparée à Alcatel-Lucent Bell Labs et à l'INRIA Sophia Antipolis

soutenue le 12 décembre 2014

Jury :

Rapporteurs :	Pr. John Breslin	-	National University of Ireland
	Pr. Guy Melançon	-	Université de Bordeaux
	Dr. Harald Sack	-	Universität Potsdam
Examinateur :	Dr. Johan Montagnat	-	CNRS, Nice Sophia-Antipolis
Directeur :	Dr. Fabien Gandon	-	INRIA Sophia Antipolis
Invité:	Johann Daigremont	-	Alcatel-Lucent Bell Labs

Á Évelyne et Olivier

Acknowledgements

Merci à Myriam Ribière et Fabien Gandon qui m'ont énormément appris, inspiré, épaulé, au-delà du professionnel,

Merci à mes anciens stagiaires, que j'ai eu le privilège d'encadrer. Merci à Damien Legrand grâce à qui Discovery Hub existe aujourd'hui, et grâce à qui mes travaux ont pris un virage décisif vers la recherche exploratoire. Merci à Émilie Palagi qui m'a accompagné durant cette dernière année et qui a obtenu de précieux résultats d'évaluation donnant à la thèse sa complétude,

Merci à mes amis et collègues Sameh Ben Fredj, Adrien Joly et Evangelos Kalampokis pour leur écoute et leurs conseils,

Merci à Olivier Corby pour son attention, son aide infaillible et pour les nombreuses possibilités techniques qu'il a ouvertes. Merci à Gessica Puri, Alain Giboin et Florentin Rodio pour m'avoir permis de valider scientifiquement mon travail via leur expertise et leur participation active,

Merci à Christine Foggia, Delphine Izanic, Fabienne Labrosse, Elisabeth Leloup et Xavier Andrieu pour leur assistance très appréciée,

Merci à mes collègues de Wimmics: Catherine Faron-Zucker, Elena Cabrio, Serena Villata, Oumy Seye, Michel Buffa, Luca Costabello, Julien Cojan, Guillaume Érétéo, Amine Hallili, Rakeb Hasan, Maxime Lefrançois et Zide Meng.

Merci à mes collègues des équipes SocialComm, Multimédia et MathDyn d'Alcatel Lucent Bell Labs: Johan Stan, Jérôme Picault, Johann Daigremont, Yann Gasté, Karim Hebbar, Loretta Maag, Olivier Martinot, Julien Robinson, Lionel Natarianni, Patrick Legrand, Bruno Legat, Olivier Durécu, Sylvain Squedin, Philippe Jacquet, Alonso Da Silva, Gérard Burnside, Dimitrios Milioris, Lamine Lamali, Amira Alloum, The Dang et Olivier Leclerc,

Merci à Caroline, Justine, Isabeau, Nathalie, Rodica, Criquette, Nenette, Jaja, Paco, Cédric, Choco, Gimousse, Frailloj, Matmaiz, Nico, Samousse et Yannick,

Merci à mes amis de toujours: Anton, Bidou, Bigou, Bilou, Boudir, Chass, Garga, Gourbi, l'Américain, Roumi, Teutif, Tos, Magic Raymond et JP Marielle,

Merci à mon frère, mes parents et à ma belle et grande famille,

Merci à Claudia avant tout.

Abstract

The general topic of the thesis is web search. It focused on how to leverage the data semantics for exploratory search. Exploratory search refers to cognitive consuming search tasks that are open-ended, multi-faceted, and iterative like learning or topic investigation. Semantic data and linked data in particular offer new possibilities to solve complex search queries and information needs including exploratory search ones. In this context the linked open data cloud plays an important role by allowing advanced data processing and innovative interactions model elaboration. First, we detail a state-of-the-art review of linked data based exploratory search approaches and systems. Then we propose a linked data based exploratory search solution which is mainly based on an associative retrieval algorithm. We started from a spreading activation algorithm and proposed new diffusion formula optimized for typed graph. Starting from this formalization we proposed additional formalizations of several advanced querying modes in order to solve complex exploratory search needs. We also propose an innovative software architecture based on two paradigmatic design choices. First the results have to be computed at query-time. Second the data are consumed remotely from distant SPARQL endpoints. This allows us to reach a high level of flexibility in terms of querying and data selection. We specified, designed and evaluated the Discovery Hub web application that retrieves the results and present them in an interface optimized for exploration. We evaluate our approach thanks to several human evaluations and we open the discussion about new ways to evaluate exploratory search engines.

Keywords

exploratory search, semantic web, linked data, linked data based exploratory search system, DBpedia, semantic spreading activation, Discovery Hub, human evaluations

Résumé

Cette thèse s'intéresse à l'exploitation de la sémantique de données pour la recherche exploratoire. La recherche exploratoire se réfère à des tâches de recherche qui sont très ouvertes, avec de multiples facettes, et itératives. Les données sémantiques et les données liées en particulier, offrent de nouvelles possibilités pour répondre à des requêtes de recherche et des besoins d'information complexes. Dans ce contexte, le nuage de données ouvertes liées (LOD) joue un rôle important en permettant des traitements de données avancés et des interactions innovantes. Nous détaillons un état de l'art de la recherche exploratoire sur les données liées. Puis nous proposons un algorithme de recherche exploratoire à base de données liées basé sur une recherche associative. A partir d'un algorithme de propagation d'activation nous proposons une nouvelle formule de diffusion optimisée pour les graphes typés. Nous proposons ensuite des formalisations supplémentaires de plusieurs modes d'interrogation avancée. Nous présentons également une architecture logicielle innovante basée sur deux choix de conception paradigmatiques. D'abord, les résultats doivent être calculés à la demande. Deuxièmement, les données sont consommées à distance à partir de services SPARQL distribués. Cela nous permet d'atteindre un niveau élevé de flexibilité en termes d'interrogation et de sélection des données. L'application Discovery Hub implémente ces résultats et les présente dans une interface optimisée pour l'exploration. Nous évaluons notre approche grâce à plusieurs campagnes avec des utilisateurs et nous ouvrons le débat sur de nouvelles façons d'évaluer les moteurs de recherche exploratoires.

Mot-clés

recherche exploratoire, web sémantique, données liées, système de recherche exploratoire à base de données liées, DBpedia, activation propagation sémantique, Discovery Hub, évaluations utilisateurs

La curiosité mène à tout: parfois à écouter aux portes, parfois à découvrir l'Amérique.

José Maria Eça de Queiros

Contents

CHAPTER 1

Introduction

Contents

1.1 Enabling new search means

"*Search is only a partially solved problem*" R.W. White, [192]

The web is the largest-known searchable information/knowledge source. However, users' access to web content is a major and continually evolving research challenge due to the changing nature of the Web. The usage, technologies and content of the Web are continuously evolving at a rapid pace. In the mid-to-late 1990s, the first popular search engines appeared. At that time, they offered a new and innovative way to access the ever-increasing amount of web pages available. In 2014, search engines are still the main mechanism used to access the Web. Users' high level of satisfaction with search engines shows that they are still effective in satisfying a large range of informational needs[1]. Their popularity is also mainly due to their intuitiveness. Search engines popularized keyword-based queries that allow a user to easily express their information needs without requiring any technical knowledge, which is quite different to how structured databases are queried.

The web constantly grows, both in size and diversity. In this context building tools to make the interaction with the web more manageable becomes critical. Moreover the users' growing expectations constantly maintain the need for innovative search approaches and technologies. Satisfying the widest range of information needs is fundamental for search engines to stay competitive. The users expect them to support a high variety of information needs. As an illustration, during a year 88% of the unique queries appear only once [9]. However, several information needs and queries remain unsolved or poorly solved by the actual search

[1] http://fe01.pewinternet.org/~media/Files/Reports/2012/PIP_Search_Engine_Use_2012.pdf

engines. They can not be successfully addressed because the actual approaches are unadapted to them, e.g. popularity-based rankings (e.g. PageRank [144]). Or because they require sophisticated human-computer interactions that are currently not supported by major search applications. There is still an important room for improvement, especially for complex queries and information needs.

In this thesis we address more specifically the case of exploratory search [120]. Exploratory search refers to cognitive consuming search tasks that are open-ended, multi-faceted, and iterative like learning or topic investigation. Such tasks are often performed on an informational domain that is unknown or poorly known by the searcher, generating a high incertitude. During such explorations the users do not necessarily know the best keywords, have a changing goal and synthesize an important amount of information. The current popular search engines are unadapted to them because they notably lack of assistance in the results consultation and explanation [120]. There is consequently a need to build systems that are optimized for these specific search tasks. It is the objective of the Human Computer Information Retrieval (HCIR) research community which crosses innovative techniques from the Information Retrieval and Human Computer Interaction fields in order to achieve it.

1.2 Exploiting structured data in searching the web

"The degree of structure of the web content is the determining factor for the type of functionality that search engines can provide" Bizer and al., [19]

The search engines update regularly their algorithms, interfaces and services in order to improve the results quality and the users' experience[2]. Such improvements rely significantly on the nature and the quality of the web data. One of the most important evolutions impacting the search engines today is the increasing amount of structured data publication on the web. Today their primary material is not only a hypertext document graph as in the mid nineties. It is now completed by numerous structured data expressed through various formalisms such as microdata[3], microformat[4] or semantic web models[5]. These formalisms allow to richly describe web resources and to make them machine-understandable. The search companies like Bing[6], Google[7] and Yahoo[8] started to leverage these meaningful data to improve the results computation and presentation. They recently encouraged the publication of structured data using an open schema they designed:

[2]http://www.seomoz.org/google-algorithm-change
[3]http://www.w3.org/TR/microdata/
[4]http://microformats.org/
[5]http://www.w3.org/RDF/Metalog/docs/sw-easy
[6]http://www.bing.com/
[7]http://www.google.com
[8]http://search.yahoo.com/

schema.org[9]. This initiative goes along with the idea that data interoperability and machine-readability are crucial for search.

The biggest source of structured data available on the web today is the Linked Open Data (LOD) cloud [18]. The linked open data cloud is a web of interconnected public datasets published by various actors including research, private or governmental initiatives. It follows the principles of the web: use of HTTP, URLs, etc. The data are formatted in the Resource Description Framework (RDF) triple-based data model, which is a W3C standard[10]. The LOD can be considered as the first deployment wave and one of the main achievements of the semantic web today. The semantic web is *"an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation"*[11]. The LOD is a very large source of publicly accessible knowledge: 31 billions of triples in September 2011[12]. Among all the datasets, the RDF conversion of Wikipedia[13], DBpedia[14], is popular and largely used by the semantic researchers and developers communities. DBpedia is regularly updated and its quality increases along the versions. Several end-users applications using it as background knowledge demonstrated its maturity.

Supporting complex search tasks requires a higher understanding of the meaning of the query and data than the level of keywords [180]. The incorporation of formal semantics in search approaches is referred to as *semantic search*. Semantic search aims to enhance search by exploiting the structured semantics at various levels. The structured data are promising for supporting complex search tasks, including exploratory search ones.

1.3 Renewing knowledge exploration and discovery

"Wholly new forms of encyclopedias will appear, ready made with a mesh of associative trails running through them" Vannevar Bush, [24]

The formal semantics allow improvements in both information retrieval and in human-computer interactions, which constitute the 2 dependent faces of exploratory search systems. They allow to innovate on algorithms and interaction models in order to lower the users' cognitive load. In this thesis we propose, prototype, and evaluate an original approach for supporting exploratory search by leveraging DBpedia. Our approach is a combination of recommendation, ranking, faceted browsing and results explanation features. Starting from one or several topic(s) selected by the users the system selects and ranks a meaningful subset of related resources in the large and heterogeneous graph of DBpedia. The users can

[9] http://schema.org
[10] http://www.w3.org/RDF/
[11] http://www.w3.org/RDF/Metalog/docs/sw-easy
[12] http://linkeddata.org
[13] http://www.wikipedia.org/
[14] http://dbpedia.org/

then interact with this results set through an interface optimized for exploration. From this initial objective we explored the feasibility of a method that computes the results at query-time by fetching the data from a distant source. First it is motivated by the fact that the LOD is distributed by nature. The proposed architecture offers flexibility in the choice of the data source(s) targeted to satisfy the information need. Second, the LOD knowledge bases are changing over the time, our approach guarantees the results freshness in this context. Third, the framework aims to overtake existing approaches by enabling a range of advanced queries. The users can notably influence the results selection and ranking schemes by tuning several parameters that directly impact the algorithm behavior. The framework is demonstrated through the Discovery Hub application[15].

More particularly we investigate the 4 following research questions (i) How can we discover and rank linked resources to be explored starting from the user topic(s) of interest? ; (ii) How to address remote linked data source for this selection? (iii) How to optimize such data based exploration approach at the interaction level? (iv) How to evaluate such exploratory search systems?

1.4 Dissertation plan

State-of-the-art review. In the first part of the thesis (chapters 2 to 4), we have carried out a state of the art review in the semantic search and exploratory search domains that are relevant to address the theoretical, technical challenges we address and to position our contributions to existing work:

The chapter 2 presents exploratory search, which is an ongoing topic in the HCIR research community. It positions it in the general field of search. This chapter details the characteristics of the exploratory search tasks and the common systems functionalities. Starting from that it identifies the desired effects of exploratory search systems. It also addresses the question of the evaluation of such systems, which is recognized as difficult.

The chapter 3 is dedicated to semantic search which is an active research field in the actual context of structured data proliferation. It presents the semantic web as well as the linked data initiative. It also details existing important semantic search systems in the research and the industry, where major initiatives recently appeared.

The chapter 4 is the prolongation of the two previous chapters and presents a state-of-the-art review of the systems allowing exploration based on linked data. It details the approaches leveraging linked data sources to help the users discovering new information and knowledge. Linked data browsers, recommenders systems and exploratory search engines are specifically reviewed. Finally it gives an overview by discussing the evolution of research, its outcomes in terms of

[15] http://discoveryhub.co

human-computer interactions and information retrieval. It also identifies the opportunities that exist today to enhance the systems.

Contributions. In the second part of the thesis (chapters 5 to 8), we present our contribution relying on the lacks and possibilities identified in the first part. We propose, implement and evaluate a linked data based exploratory search framework and web application: Discovery Hub.

The chapter 5 proposes first a novel algorithm that selects and ranks resources of interest starting from the users' inputs. This algorithm is an original adaptation of spreading activation. It is specifically designed to be applied on semantically heterogeneous graphs. It leverages the graph semantics in order to retrieve results that are strongly related and similar to the users' topics of interest. This chapter also presents several advanced query formulas that are variants of the core one. These variants enable a multi-perspective exploration that goes beyond the state-of-the-art of linked data algorithm-based exploratory search. This chapter addresses the formal, algorithmic level.

The chapter 6 presents the execution of the algorithm on DBpedia. Our approach overtakes the state-of-the-art techniques in terms of data selection and querying flexibility. Indeed the results are computed at query-time from distant linked datasets. It allows to select the SPARQL endpoint used to process the query and to tune the algorithm parameters before executing it. To do so we couple the spreading activation algorithm to a linked data import procedure. In other words the algorithm is applied locally on a finely selected sub-graph that is incrementally imported on run-time from a targeted SPARQL endpoint. The chapter also presents the multiple analyses of the algorithm behavior we did in order to calibrate its main parameters. This chapter addresses the data processing level.

The chapter 7 presents the Discovery Hub application, which implements the framework proposed in chapter 6, and describes its interaction model. It motivates our design choices and shows how the semantics empower the interactions in an exploratory search context. Its design is extensively discussed regarding the first Discovery Hub prototype, the existing guidelines and the desired effect of exploratory search systems. Discovery Hub is one of the most mature and complete linked data based exploratory search system at the time of writing. It implements a variety of functionalities which support the heterogeneous searcher needs e.g. faceted browsing, multiple explanations, memorization features. It is currently in a stable state of production. This chapter addresses the human-computer interaction level.

The chapter 8 presents the Discovery Hub evaluations. It details the novel protocols and results of several users' experimentations aiming to evaluate the different algorithm variants. Contrary to other initiatives in the field our evaluations relied only on users' judgment, using more than 5.000 ratings. The chapter also presents an ongoing reflection which aims to overtake the actual evaluation approaches. The corresponding protocol has notably the objective

to compare the exploratory search systems on a fair basis. The preliminary results we obtained are presented. This chapter aims to validate our contributions and to open the reflection about the evaluation of exploratory search systems.

Finally **the chapter 9** concludes and opens the reflexion by proposing short-term improvements as well as new research perspectives for linked data based exploratory search systems.

CHAPTER 2

Exploratory search

Contents

2.1 Introduction

Searching, sharing and understanding information are basic human psychological needs. Humans are explorers by nature and strongly need to integrate with and learn about the world [194]. Today information infrastructures, including the web, are massive, sophisticated and intensively used in our societies. Information retrieval has been one of the most prolific research fields ever in computer sciences. The web can be seen as the biggest document and data base with search functionalities. Since their appearance in the mid-nineties web-scale search engines revolutionized information access and impacted billions of users in their professional and personal activities. Major search engines are engineering master-pieces that give access to billions of web pages and data. They are constantly renewed to satisfy the users increasing expectations. Moreover the competition between the major search companies is intense and catalyzes the industrial research.

Web search engines success proves that they are efficient tools. Nevertheless they are more efficient when the user has a precise information need (e.g. looking up for Claude Monet's birth-date[1]), than when the user has a vague and open

[1] *Claude Monet* is a recurrent example for Discovery Hub. It was initially chosen for a screencast presented during the Semanticpedia presentation which was strongly related to culture. *Claude Monet* was the first art-related query entered by a user in Discovery Hub

information need e.g. discovering impressionism. Such information tasks are referred in the literature as cases of exploratory search. A new generation of search engines is currently under research by a community composed of information retrieval and human-computer interaction experts. These applications aim to solve exploratory search needs thanks to supportive functionalities. The research about such systems is still in its infancy and faces several important difficulties. One of them is to characterize what an exploratory search task is in order to properly design the systems that will support it. The second one concerns the evaluation of these systems which is difficult regarding the task complexity and the users' high engagement in the search process.

In this chapter we will review (2a) the evolution of the practices and technologies of web search as well as the limits it faces today, (2b) a taxonomy of search activities and the positioning of exploratory search in it, (2c) the characteristics of exploratory search tasks as defined in the literature, (2d) the corresponding applications functionalities and finally (2e) their evaluation.

2.2 Search

2.2.1 Emergence

Search is one of the most popular application on the web and an application with significant room for improvement. In the nineties the web was growing exponentially and the users were confronted to an overwhelming amount of information that was increasingly difficult to access. The web pages were mainly organized in form of hierarchical directories. Yahoo (*"Yet Another Hierarchical Officious Oracle"*) was one of the most popular one, see figure 2.1. These directories had the form of menus. At that time menus were the most frequent computer interaction style for selection and browsing and constituted therefore a natural inspiration to propose an organization of the web.

Web scale search engines were revolutionized by the linked-analysis algorithms. These algorithms relied on the idea that the hypertext links constitute clues about human judgment. The fact that someone links his website to another has an informational value. Thus, the web graph structure can be analyzed to determine which content are the most informative and/or popular. In 1998 Jon Kleinberg proposed the HITS (Hyperlink-Induced Topic Search) algorithm in order to rank the web pages [95]. The algorithm identifies hub pages (that redirect to sets of interesting pages) and authoritative sources (that constitute results of interest). The HITS algorithm is conceptually powerful but leads to critical implementation issues at web-scale. These issues include the fact that it processes a targeted subgraph and not the whole web graph, it is slow and vulnerable to spamming [99]. Later in 1998 Sergei Brin and Larry Page presented the PageRank algorithm in a paper titled *"Bringing order to the web"* [144]. The PageRank algorithm was inspired by the academic citations analysis techniques used in research. A page has a high rank if the sum of the ranks of its back-links is high. When it is applied on a graph the

Figure 2.1: The Yahoo directory website in 1995

rankings propagate iteratively through the links. Contrary to HITS the PageRank is query-independent. It was able to rank the whole graph of the web: 150 million pages and 1.7 billion links in 1998 at as mentioned in the article. In the paper the authors also introduced their first PageRank implementation in a keyword-query search engine: Google. The Google search engine has constantly evolved since the Brin and Page's article, it now uses a massive amounts of heuristics that are mainly undisclosed. For a detailed comparison between HITS and PageRank algorithm the reader might refer to [99].

2.2.2 Popularity

Being able to index and rank the whole web was a major technical achievement in the mid-nineties. The new interaction mode proposed by the search engines was another major key factor of success. Indeed search engines became very popular because they allowed the users to express their information needs in the form of keywords. This query paradigm is instantly accessible to everybody. It is intuitive and fast. Before, search was mainly accessible to database experts in enterprise applications. It was internal, limited to a closed (often business-related) domain, required technical skills and was not available at web-scale. The keyword search paradigm is independent from the data models, structures and query languages of the data/information collection searched. As mentioned in [180]: *"search is rather understood as an end-user oriented paradigm that is based on intuitive interfaces and access mechanisms"*. The first popular search application was Altavista[2] which answered 13 million queries a day in 1998 [10]. The same year the Google company was founded and became rapidly the leader of the search market. Today, search

[2]http://www.altavista.com/

is still the main way to access information on the web[3]. At the time of writing Google is the most popular application on the web according to Alexa[4]. New markets emerged from search engines popularity including search engine optimization and words-based advertising e.g. Google Adwords[5].

2.2.3 Limits and opportunities

The users are now familiar with search engines. They expect them to support increasingly complex search tasks. There is a constant need for innovative search approaches and technologies as the web becomes more and more complex and the users' expectations increase. As mentioned in [194] the users' expectations about search systems are exceeding the systems' current capabilities. The fact that the average number of keywords per query is growing[6] may confirm that the users expect the search engines to support increasingly complex information needs. It might also be an indication about the increasing users' search expertise. Even if search engines are very efficient tools for a majority of common information needs they perform poorly on several types of hard queries including ambiguous, imprecise or overly precise queries. Advanced query syntax like booleans can help but it is not used by the majority of users.

Today search engines are unadapted to solve some complex information needs. First they are limited by their keyword querying paradigm. Indeed all the real-world information needs can be hardly covered with small sets of keywords. Second, the search companies based their success on the simplicity of use of their products. The simplicity of the interface and interactions acts as an informational bottleneck for complex search tasks. Third, the investment and research in favor of their retrieval speed, over billions of web pages, was done by sacrificing potential relevance by using models, algorithms and heuristics that limit the complexity in order to speed up the processing [178]. Fourth, we observe today a *filter bubble* effect: the over-personalization of search results tends to retrieve always the same kind of results to the users and to limits their exploration and discovery possibilities [146].

2.3 Exploratory search

In 2006, Gary Marchionini introduced, defined and popularized the term *exploratory search* in the seminal paper *"exploratory search: from finding to understanding"* [120]. The topic was not new. Prior to it a large amount of papers were already focused on similar search tasks. According to [194] the main related topics were

[3] http://fe01.pewinternet.org/~media/Files/Reports/2012/PIP_Search_Engine_Use_2012.pdf
[4] http://www.alexa.com/siteinfo/google.com
[5] https://adwords.google.com
[6] http://www.hitwise.com/index.php/us/about-us/press-center/press-releases/2009/google-searches-apr-09/

information foraging, berry-picking search approach, sense-making, information-seeking and cognitive information-retrieval. A list of relevant references is available in [194]. The Gary Marchionini publication inspired many researchers and a community centered on the topic emerged. It is important to notice that at the current state of research the definition of exploratory search is unstable and is still shaped by the ongoing research. We propose the following resuming definition: exploratory search is an increasingly directed search activity with expectations of discoveries.

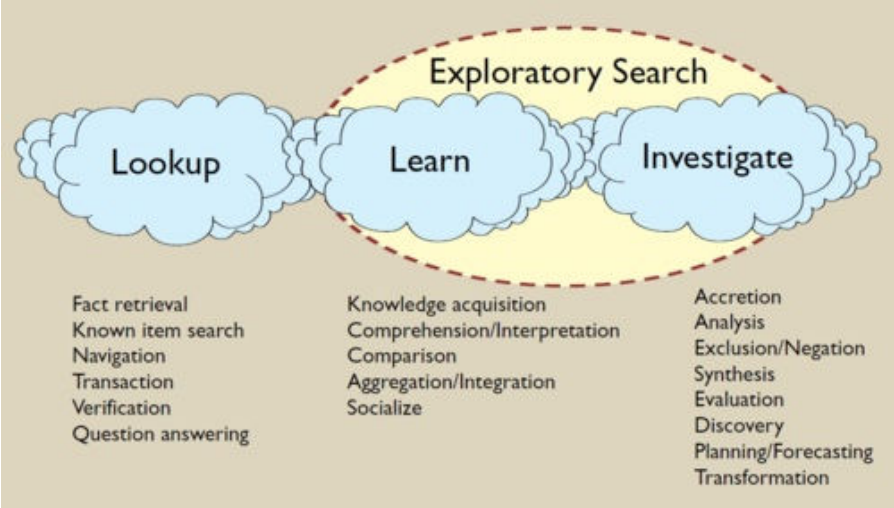

Figure 2.2: Taxonomy of search tasks proposed by Gary Marchionini in [120]

2.3.1 Definition

In his paper Gary Marchionini proposed a taxonomy of search activities. In order to characterize what are exploratory search tasks he put it in perspective with lookup tasks, see Figure 2.2. Lookup task, defined hereafter, are simpler and more familiar to the users:

Lookup tasks refer to search tasks when the users look for something in particular, having a well-defined information need. They correspond to queries that aim to retrieve a discrete result e.g. a website, a statement, a name, a phone number, a file. The lookup queries are performed in case of known-item search (including navigational query seeking a single website), fact retrieval, close-ended questions-answering and verification. In [120] Gary Marchionini precises that *lookup tasks are usually suited to analytical search strategies that begin with carefully specified queries and yield precise results with minimal need for result set examination and item comparison*.

The actual major search engines are very efficient for lookup tasks due to their keyword-based query paradigm. Indeed during lookup tasks the users' information need is precise, consequently they can easily formulate effective keywords. The latter are easy to match with the search engines index where the documents

collection is stored in the form of a bag of keywords. Consequently the systems often quickly answer such information need. The objective is to reach the desired result as fast as possible. In this spirit, the Google *I am feeling lucky* button[7] directly displays the first result, skipping the results list page, see figure 2.3. The knowledge panels, see example on Figure 2.4, proposed by the major search engines today also reflect the tendency to retrieve direct answers that does not require a results list examination. The 3 major search engines released their own knowledge panrls: the Bing Snapshot[8], the Google[9] and Yahoo[10] Knowledge panels. These functionalities and their underlying technologies are extensively described in chapter 3 and 4 of this thesis.

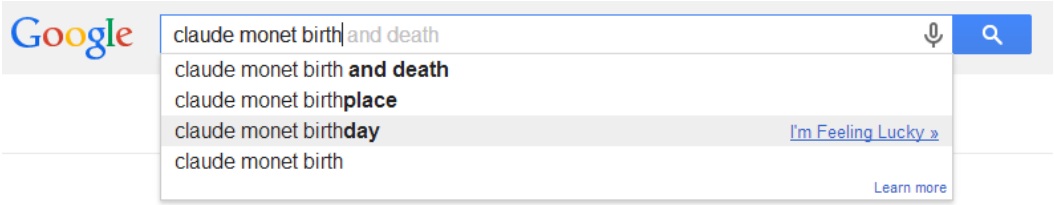

Figure 2.3: An example of actual search engines' lookup optimization's orientation: the *I am feeling lucky* button skips the results directly opens the first result

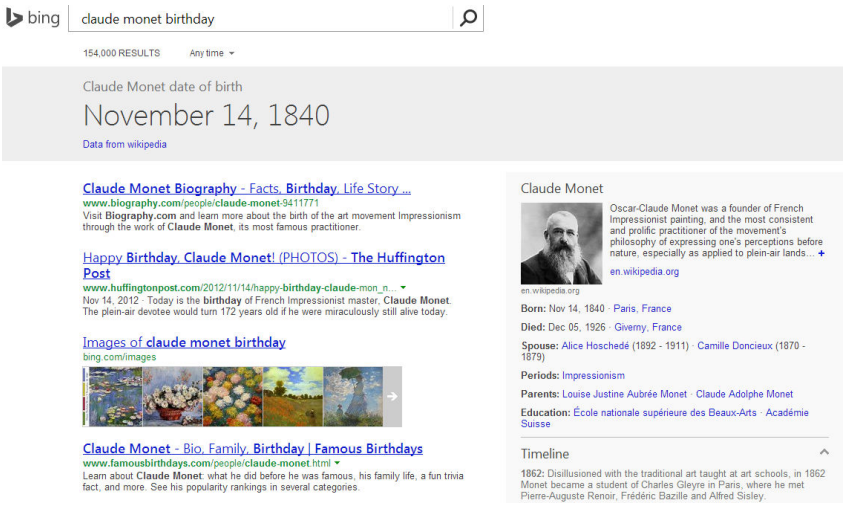

Figure 2.4: An example of lookup query that retrieves the birthday of Claude Monet using Bing

Exploratory search refers to cognitive-consuming search tasks like learning or investigation. The terms learning and investigation are employed in the broad

[7] https://en.wikipedia.org/wiki/Google_Search#.22I.27m_Feeling_Lucky.22
[8] http://www.bing.com/blogs/site_blogs/b/search/archive/2013/03/21/satorii.aspx
[9] http://www.google.com/insidesearch/features/search/knowledge.html
[10] http://searchengineland.com/yahoo-testing-knowledge-graph-lookalike-search-results-189505

sense. They can occur in educational, professional or personal contexts. Generally speaking the objective in this case is to create a knowledge product or to shape an action in a complex and rich informational context [194]. In 2006, R.W. White proposed this definition [193]: *Exploratory search can be used to describe an information-seeking problem context that is open-ended, persistent, and multi-faceted; and to describe information-seeking processes that are opportunistic, iterative, and multi-tactical. In the first sense, exploratory search is commonly used in scientific discovery, learning, and decision making contexts. In the second sense, exploratory tactics are used in all manner of information seeking and reflect seeker preferences and experience as much as the goal*. [81] proposed the following city-exploration analogy: *this is similar to how one might actually want to walk around a city to understand its layout and enjoy its culture rather than taking a quick taxi ride to a specific destination*. It is noticeable that lookup and exploratory search tasks overlap. When learning a user might perform several lookup tasks to clarify some precise points for instance. Exploratory search is composed of two main activities: exploratory browsing and focused searching [194]. [106] defines browsing as *"a movement in a connected space"*. Exploratory browsing corresponds to an undirected activity that aims to better define the information need and to raise the understanding of the information space. Focused searching is more directed and occurs when the information need is clearer. More precisely it includes query refining, fine results consultations and comparisons. The inter-dependency between the evolving precision of the information need and the executed search activities is illustrated in Figure 2.5.

2.3.2 Tasks

There is an effort in the research community to observe, understand and model exploratory search behaviors and tasks [45] [195]. The first objective is to raise the level of shared understanding on this topic among the research community. A lot of researchers use different (but close) exploratory search definitions, leading to confusion and approximation. The second objective is to propose guidelines for designing exploratory search functionalities and applications. Last but not least, there is a need to agree on best practices for evaluations tasks and scenarios design. Using similar evaluations will result in better comparisons baselines and consequently better findings and systems improvements. It requires first solid definitions.

At the time exploratory search is often defined by its key task attributes. We merged and synthesized the characteristics of exploratory search tasks that are often cited in literature (including in particular [195] and [194]) and present them hereafter:

- The **goal is to learn or investigate**. This point, present in the fundamental paper of Gary Marchionini, is well cited in the literature. Common contexts in which exploratory search occur are educational, academic, work and leisure ones. Sometimes the users are motivated by curiosity and have a very vague

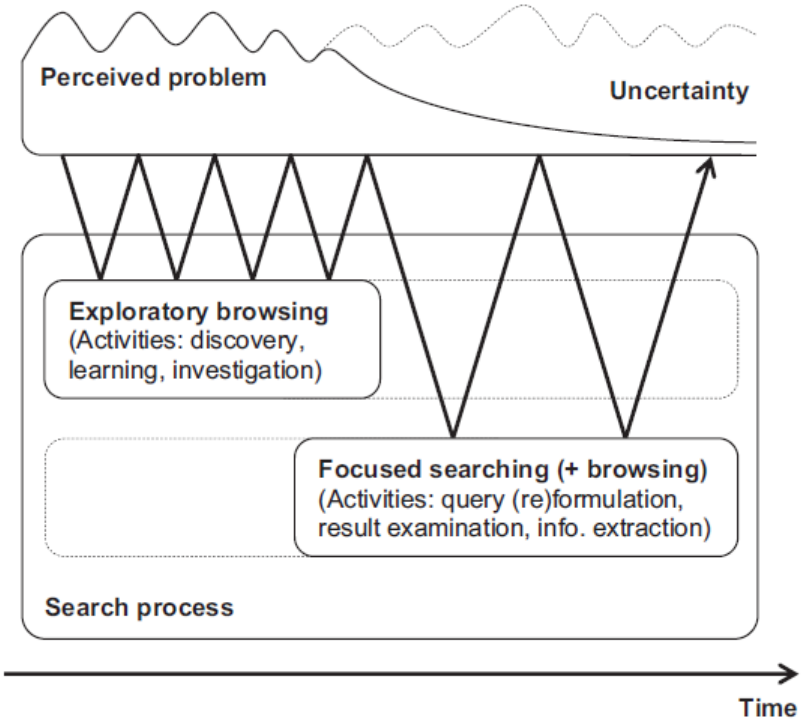

Figure 2.5: Interactions between the users' information need precision and the executed exploratory search activities, taken from [194]

idea of what they are expecting. In this case the curiosity is the principle motivational factor. The exploration is not driven by a defined information need. This case was referred to as *casual-leisure information behaviors* and is related to personal, non-professional activities for *"achieving a hedonistic goal"* [197]. These explorations are particularly susceptible to engender discoveries. For instance the users may open their favorite canals (e.g. forums) with the motivation to encounter interesting content without specific expectancies.

- The **search tasks are general rather than specific**. The user have consequently an important latitude in the manner they execute the search activity. The information goals are not well-defined (*"vague"*, *"amorphous"*, *"ill-defined"*, *"lacking of structure"*) and there is not a concrete answer to find. The absence of focus in the search domain and objective leads to an undirected search process. Nevertheless during the exploration the users progressively build a mental answer-framework that helps them to focus the exploration on specific aspects of the topic.

- The **search task is open-ended** and can be distributed over the time. It can last for a certain period as it can be motivated by a long-lasting and/or evolving interests. Consequently the exploration occurs during multiple search

iterations and sessions. [178] introduced the *"slow search"* practice which means *"a class of search where traditional speed requirements are relaxed in favor of a high quality search experience"*. In the case of rich search experience, including exploratory search, the time is less pressing. A trade-off exists between the results quality and the retrieval speed. According to Microsoft Research [47], more than 50% of search occurs during multi-queries sessions exceeding 30 minutes. Search queries are often not performed in isolation. Such findings tend to confirm that an important part of search activities are exploratory ones.

- Exploratory search is **cognitive-consuming and related to non-trivial information needs**. Contrary to lookup search tasks, that often take the form of *one-shot queries*, there is a continuous interaction between the user and the system during the exploration. The users interact with results having different types (e.g. data, texts, videos) through a variety of interaction modes. Usually they engage the exploration by performing an imprecise query that retrieves a large set of potentially relevant results. This interaction style is known under the name of *orienteering* [141]. The users first naively traverse the information space [194]. In a second time they spend more time to finely consult the results in order to rise their understanding and mental representation of the topic. Exploration search is always correlated with a plurality of sense-making [94] [91] activities such as comparison, analysis and evaluation of the results. One definition of sense-making is "*how people make sense out of their experience in the world*" [46]. It is conceptualized as a two-way process of fitting data into a frame (which is roughly a mental model) and fitting a frame around the data. The complexity of the information need leads also to a multiplication of searching and browsing sub-tasks that make sense together. The users perform an important amount of small and diverse cognitively-manageable tasks in order to reach the exploratory search objective incrementally. These profusion and heterogeneity of interactions are sometimes referred to as a *"procedural complexity"* [195]. According to Gary Marchionini the users' high investments in time, effort and creativity are rewarded by a deep knowledge [120].

- There is a **varying level of uncertainty during exploratory search**. This uncertainty can concern the search goal, the manner to achieve the goal, the domain investigated, the results and even the search systems' use. Exploratory search is a case of *"weak problem solving"* [139]: the users lack prior domain knowledge and progress through unsystematic steps through the information space [194]. The uncertainty generally decreases along the search as the familiarity with the topic explored increases. However the uncertainty can also increase when the users discover new information that disturb the mental models they are currently building. Such evolving degrees of knowledge and incertitude suppose different expectations in the systems. If the users have a limited knowledge of the information domain explored they

will have difficulties to formulate the queries and to interpret the results. They will need the system to be supportive regarding these aspects. When the users are familiar with the domain explored they are more interested by gaining deeper knowledge about it e.g. identifying knowledge gaps or new, unexpected points of view.

- Exploratory search **information need is evolving**. As the information need is fuzzy it is likely to evolve during the exploration being influenced by the informational context. As mention before the users show an orienteering behavior in a first time. They retrieve first some results and facts that inspire new strategies e.g. new queries, new navigation paths, etc. The information exposure and knowledge acquisition that occur during exploratory search tasks lead to changes in the users' objectives along the time. Moreover their perception of relevance also evolves along the time [194]. External events can also impact the information objective during long-lasting search e.g. news. The users often have also an opportunistic behavior. They explore several sub-sets of the information space through orienteering and "*pivot*" several times, see figure 2.6. Having initial imprecise expectations is characteristic of exploratory search. It consequently leads to many discoveries of unknown information and associations.

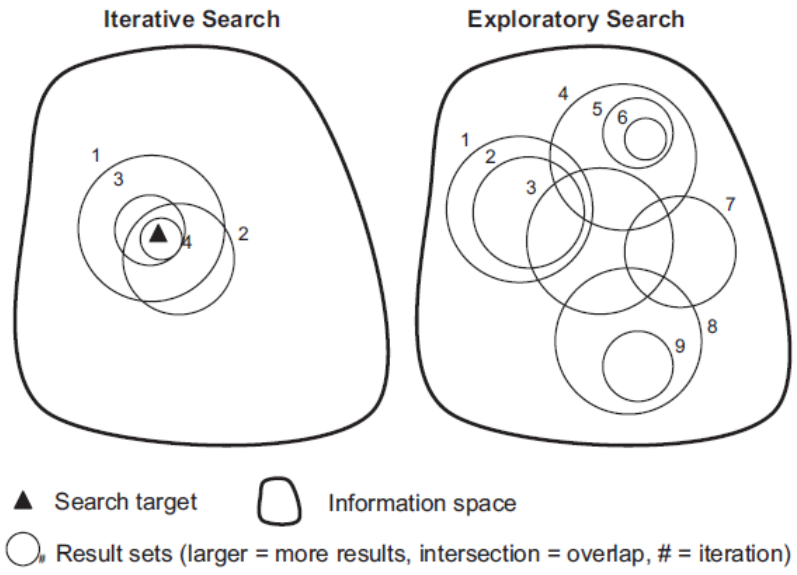

Figure 2.6: Comparison between iterative search used during lookup and exploratory search strategies, taken from [194]

Actual major search engines are not adapted to exploratory search. It is impossible to satisfy complex information needs, especially vague ones, by representing them in few keywords processed in a *query-response* mode. Today when the users perform exploratory search on actual search engines they manipulate iteratively

an evolving set of keywords. They have to synthesize an important amount of information coming from a changing result space without support from the system. The search engines do not help the users in the results exploration as they are optimized for (*"one-shot query approach"*). The users have to rely on their own search strategies. It leads to important cognitive load: *"the human user's brain is the fundamental platform for information integration"* [21]. To sum up when used for exploratory search the actual popular search engines act like information bottlenecks. They force the users to adopt compensation strategies that are very cognitive consuming.

"Rather than viewing the search problem as matching queries and documents for the purpose of ranking, interactive information retrieval views the search problem from the vantage of an active human with information needs, information skills, powerful digital library resources situated in global and locally connected communities - all of which evolve over time". Gary Marchionini, 2006

For these reasons Gary Marchionini encouraged to complete existing solutions with functionalities and cognitively optimized for exploratory search tasks. Such systems have to ease memorization, the understanding, the emergence and refinements of new mental models. The information retrieval, human computer interaction and associated disciplines experts collaborated on this objective, see figure 2.7 Gary Marchionini also coined in 2004 the term *Human-Computer Information Retrieval* (HCIR) to stress the importance of the HCI apects to overtake the actual search systems limitations [121] by increasing the level of interactions with search systems. Since 2007 the HCIR community organizes an eponymous workshop[11][12]. Exploratory search is one of its major topic.

2.3.3 Systems

Matching systems and users goals. The majority of the web search applications are based on the classic information retrieval *query-response* model that have been extensively used since the apparition of computers [12], see figure 2.8. The information need is captured in the form on a query and treated through a *one-shot* processing. Several queries are successively performed if the need is not satisfied in the first place (trial-and-error tactic). This model does not capture the complexity of the users' interactions during a complex search session query refining, browsing, results comparison, etc. For this reason this model has been regularly criticized since the eighties [194]. To overtake the *lookup-oriented* query paradigm it suggests, Gary Marchionini proposed in [122] an information-seeking process model aiming to better represent the interactions between the users and the applications. Observing this model on figure 2.9 it appears that systems based on it

[11] https://sites.google.com/site/hcirworkshop/
[12] http://hcil2.cs.umd.edu/trs/2006-18/2006-18.htm

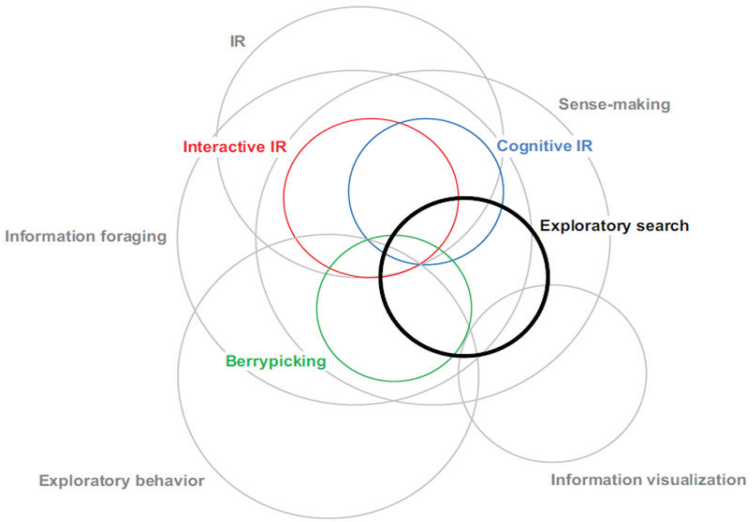

Figure 2.7: Venn diagram positioning exploratory search relative to others disciplines, taken from [194]

are far more dialogical. They support multiple and heterogeneous man-machine interactions along the different exploratory search steps.

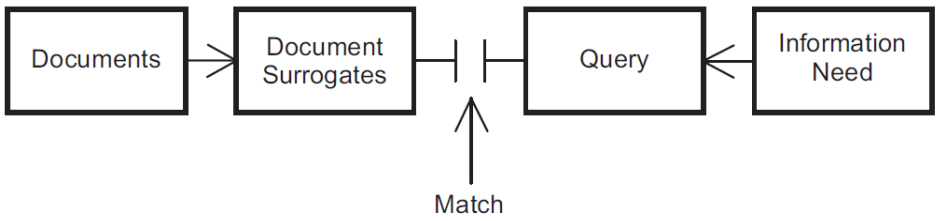

Figure 2.8: Classic information retrieval model as proposed in [12]

Exploratory search systems and functionalities are built specifically to help the users during cognitive consuming search tasks. Their objective is to satisfy information needs when the searcher has a limited knowledge of his goal, domain or search system [192]. Consequently they aim to go beyond the widespread *query-response paradigm* [120] and to propose a more interactive experience. They do not retrieve precise and immediately satisfying answers but support the users all along the exploration through a variety of supportive features. The emphasis is put on the search experience and process rather than just on the results: *"the path is the destination"*[188].

Gary Marchionini stated *"the search system designer aims to bring people more directly into the search process through highly interactive user interfaces that continuously engage human control over the information seeking process"* [120]. They guide the users all along the navigation through an interactive search process. They assist them in building complex queries from several atomic and/or incremental operations

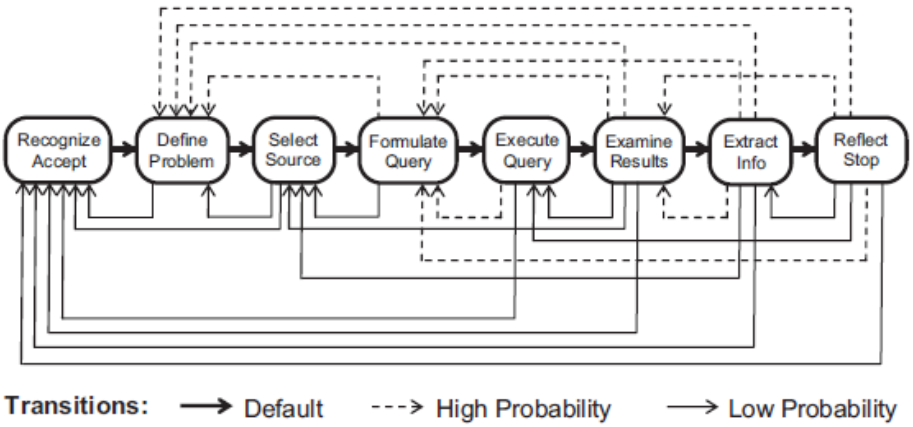

Figure 2.9: Information-seeking process model proposed in [122]

[101]. They also propose a variety of navigation trails that ease the exploration of an unknown or poorly known domain. Overall their objective is to fractionate the exploratory search tasks in series of smaller tasks that are more manageable from a cognitive point of view.

Heterogeneity of the functionalities. Existing exploratory search systems considerably vary in the functionalities they propose. We synthesize below the systems widespread features that are observable in the literature:

- **Overviews, visualization and analytics features** help the user to understand the nature of the results and the results themselves. Overview features are used in some exploratory search systems. Visualization on data are often associated to specific properties e.g. timelines for time-stamped data, maps for geotagged ones.
- **Faceted interfaces** [72] are especially popular and ease the results exploration by grouping and filtering them using their common properties. The presence of facets also helps the user to understand the nature of the results by making explicit their important characteristics. The facets constitute a powerful support for shaping the information need and the corresponding answer-framework. Exploratory search systems favor query refinement through dynamic querying interfaces rather than the multiplication of *one-shot* queries.
- **Clustering of the results** can be performed to help their understanding and consultation. It is especially useful when the system retrieve data having heterogeneous type and coming from various domains. Clustering can be an alternative to faceted browsing for organizing the content.
- The users have the choice between **multiple alternative browsing paths** through *going back-and-forth* functionalities . The risk of following different

paths is low and is perceived as such. In other words it is easy to come back to a previous step of browsing during the exploration process. As the users might have a very limited knowledge of the domain they explore multiple paths in an undirected manner. To ease the process of exploration one possibility is to give indications on the effects of an interaction before the user effectively engage it. An example is the use of numeric volume indicators [196] (NVIs) and previews cues. In faceted search context the NVIs indicate to the users how many results will be displayed if a facet value is selected. It helps to decide which facets to activate/deactivate in order to restrain or augment the focus. Some systems present directly the amount of results corresponding to the facet value [204], others systems uses more graphical presentation [26]. Preview cues give an idea about the content associated to the previewed facet through summary, images and sounds for instance.

- Exploratory search systems can offer functionalities that **narrow and broaden the query**. Such functionalities help the users to shape their information need. They propose suggestions that refine or modify the users' original scope of search through auto-completion and recommendations of associated query-terms/resources[188]. It is especially helpful when the users have a fuzzy information need or poor knowledge of the domain leading to difficulties in query formulation. Recent researches aimed to model the user intent for exploratory search purposes in order to propose adaptive interfaces[161].
- Surprises, discoveries and novelty can be enforced are part of exploratory search. Some of these systems voluntarily influence their results to favor **serendipitous discoveries** [5].
- The exploratory search systems often offer **memory-features** such as in-session or account-related history, task resuming and information gathering capabilities. It is especially helpful for long-lasting exploration. By tracking the sequences of interactions they free the users memory and allow them to concentrate on the exploration task.

We propose a summary of how exploratory search tasks are supported by the systems functionalities in the following diagram, see Figure 4.34. The *exploratory search tasks characteristics* on the left are summarized from [194]. We derived from these characteristics a list of *desired effects of the systems* and linked them to the previously listed widespread features. It is important to notice that the systems implement sets of features that often support selected aspects of exploratory search tasks. They might also use different functionalities to obtain the same effect. For instance a result visualization and a clustering can both offer an interesting support for understanding the informational domain. Due to resources constraints the systems designers and researchers have to make design choices and focus their efforts on one or several key functionalities. Exploratory search functionalities have all their strengths and weaknesses. Their combination in the systems through some

interaction models constitutes the core value. It is a complex alchemy.

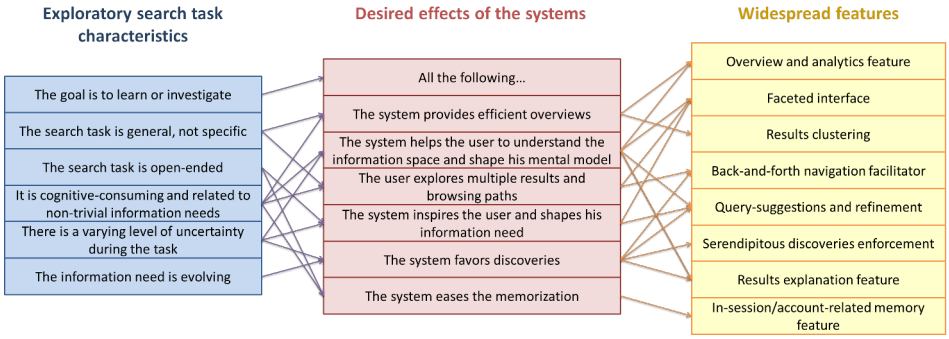

Figure 2.10: Relation between exploratory search tasks characteristics and widespread systems features

New exploratory search contexts are also investigated. This is the case for mobile exploratory search systems as in [200] and [138]. The size of the screen is especially challenging for designing the interaction model. Collaborative exploratory search tasks, algorithms and interfaces have also been investigated e.g. [136], [150], [58]. The exploration of a topic by several users can offer a more complete view on it by bringing together different perspectives, experiences, expertise and vocabulary [194].

Interface design. Due to the complexity of the interactions they propose designing efficient exploratory search system is a challenge. There is a high tension between the variety of functionalities proposed and the clarity in design [198]. Thus human-computer interaction research is fundamental to enhance such interaction models and interfaces. Theyr have to be intuitive in order to propose systems that are easily usable by non-expert. One objective of the design is to reduce the users' cognitive load in the difficult task of exploration. The systems have to free the cognitive resources of the users at the maximum (e.g. their memory) to allow them to focus on high value exploratory search sub-tasks such as knowledge accretion, analysis, results comparison, etc. There is also a need to favor a more active human engagement in the search process. Some guidelines for designing exploratory search systems can be found in the literature. The following ones, elaborated for faceted browsing in an exploratory search context are summarized from [201]:

- **Maintain keyword search**: keyword search should be maintained in faceted interfaces. When both are available they are used evenly [199].
- **Think about different users**: it is important to support various search strategies to satisfy different types of users. Some of them might be relatively confident of the outcomes they want whereas others have strictly no idea of what they can find, expect.
- **Show information in context**: the visibility of facets constitutes interesting contextual cues. They can be used passively to understand the domain.

- **Include sorting and filtering**: the ability to arrange the results is a key feature of exploratory search systems.
- **Facilitate information gathering**: the ability to collect, bookmark, keep traces of information found is important for long-lasting explorations
- **Offer previews**: previewing the effect of the interactions is helpful, especially when the user has a vague knowledge about the domain he is searching.

Industrial deployment. Major web players started to deploy basic exploratory search systems and functionalities. Google launched in 2012 the knowledge panel (*"explore your search"*, *"things not strings"*[13]). This functionality takes advantage of the Google Knowledge Graph semantic network. It assists and inspires the user in the exploration of a topic of interest by notably retrieving a list of related items, see figure 2.11. In 2013 Bing[14] and Yahoo [20] released similar knowledge panels, also based on semantic graphs. The Facebook Graph Search functionality[15] can be considered as an exploratory search functionality to some extent. It combines autocompletion and faceted filtering to explore the social network graph. It exposes unexpected information that cannot be accessed by browsing. One of the most interesting application in the field of exploratory search systems today is Stumble Upon [16], see figure 2.12. Stumble Upon is a web content discovery engine that relies on social profiling mechanisms. It recommends web pages regarding corresponding to its users interests. The recommendations are sorted by categories e.g. *internet, design, music*. Stumble Upon is popular as it was ranked the 152th most popular website in March 2014 according to Alexa[17]. TasteKid[18] is a discovery engine that suggests musics, movies, television shows, books, authors and games starting from users' tastes. The method used for computation is proprietary and unknown but makes use of Freebase[19] [20]. The underlying recommendation engine appears to be powerful and is able to perform cross-type recommendations (e.g. books from a movie) as well as multi-inputs ones (e.g. video games from a book and a movie). Unfortunately it does not provide explanations to the users about the recommendations and appears as a *black-box*. Similarkind[21] offers movies, television shows, musical artists, books and video games recommendations. Contrary to Tastekid it does not provide multi-inputs nor multi-types recommendations. It does not give any explanation about the results retrieved neither. Similarkind is based on the Freebase knowledge base and on Wikipedia according to its website.

[13] http://www.google.com/insidesearch/features/search/knowledge.html
[14] http://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/
[15] https://www.facebook.com/about/graphsearch
[16] http://www.stumbleupon.com/
[17] http://www.alexa.com/siteinfo/stumbleupon.com
[18] http://www.tastekid.com/
[19] https://www.freebase.com/
[20] http://blog.tastekid.com/post/45346321962/two-weeks-of-work-at-tastekid/
[21] http://www.similarkind.com/

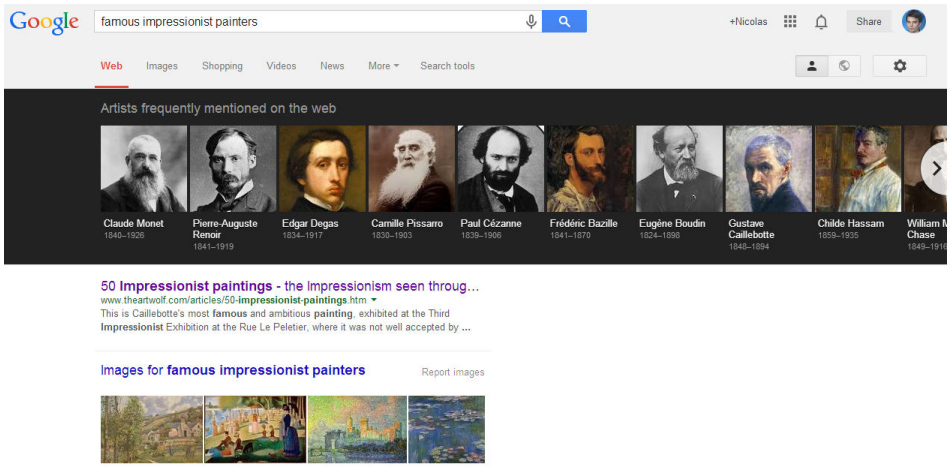

Figure 2.11: A list of impressionist painters retrieved by Google

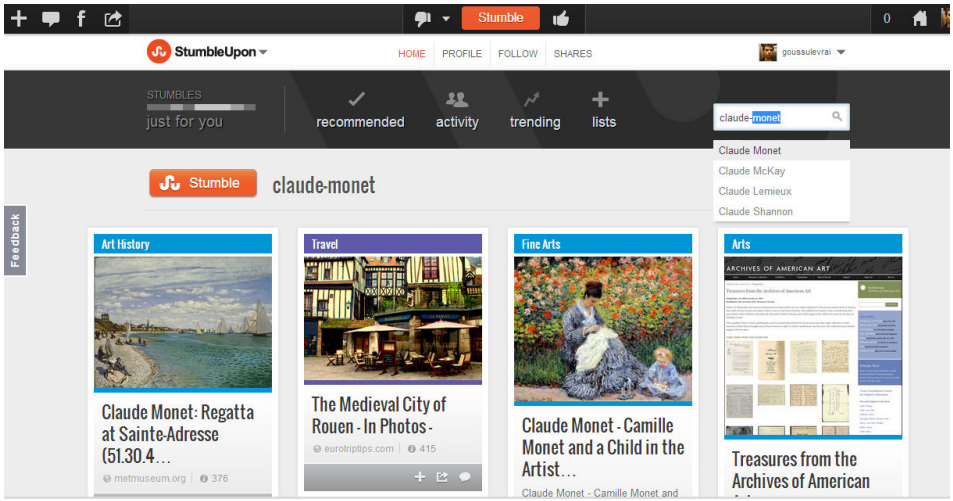

Figure 2.12: Exploration of the Claude Monet interest with StumbleUpon

2.3.4 Evaluation

The exploratory search systems evaluation is recognized by the HCIR community as difficult. The reasons are multiple. First, the traditional measures used in information retrieval are mainly focused on the results ranking. **Precision** is *"the fraction of retrieved documents that are relevant to the query"* and **recall** is *"the fraction of the documents that are relevant to the query that are successfully retrieved"* [119]. Exploratory search systems often favor the recall more than the precision [120]. Indeed it is more interesting to present a vast amount of potentially interesting and useful results to be explored rather than presenting only the most relevant ones. Second the task completion time is not a valuable metric. As the systems put the emphasis on the exploration process the speed of a task completion can not constitute a valuable metric [178]. On the contrary some evaluation protocols assume

that efficient exploratory search systems encourage their users to explore during a long time [25]. Third, the comparison of exploratory search systems among themselves is also difficult because they have very diverse features and design[201]. For instance [169] reported 14 dimensions for categorized overviews that have been used in the literature (till 2009). This diversity leads to a high amount of independent design variables that are difficult to compare. Fourth the users' fatigue inherent to exploratory search and the duration of the tasks prevent from evaluating a large amount of queries. Evaluation of such systems remains today an active research topic. The elaboration of more precise research questions, protocols, tests collections and tasks is needed. They will lead to more objective studies, better comparison that will then result in significant systems' improvements. Today many candidate methods are tested and often lead to results that are difficult to compare among themselves.

Task-based scenarios are sometimes used to evaluate exploratory search systems. To be realistic the users have to be in a situation where they lack knowledge and have a fuzzy information need. Moreover the task has to be sufficiently complex to require browsing and exploration and should lead at a certain point to discoveries. This is hard to design. Moreover during an exploratory search task the outcomes are very subjective: the users interpret the task, the results and their relevance. In the past works both [103], [204] and [102] proposed simulated scenarios that needed an exploratory search tasks to be satisfied. As reported in [102], based on the existing literature, the desirable characteristics of exploratory search tasks are: *"uncertainty, ambiguity, discovery, having an unfamiliar domain for the searcher, providing a low-level of specificity about how to find the information, constituting a situation that provides enough imaginative context for the participant to relate and apply the situation"*.

In [102] the authors looked at the log sessions of the system they evaluated to identify which topics were subject to exploratory search behavior. These topics engendered search sessions that notably included the use of facets. A candidate topic was for instance *"british history"*. Then they used the following template scenario to generate four exploratory search tasks: *"Imagine that you are taking a class called xxx. For this class, you need to write a paper on the topic xxx. Use the catalog to find two possible topics for your paper. Find three books for each topic"*. They also introduced two structured tasks (e.g.*"Look up background information in a book titled Firefly encyclopedia of trees"*). Introducing these two structured tasks helped the authors to verify if their task design was successful i.e if the users perceived a difference between the exploratory search tasks and the structured known-item tasks. It was a success as the participants often grouped the four exploratory search tasks together as well as the two structured search tasks. The participants also affirmed that:

- They were more familiar with the structured search tasks.
- It was less easy to accomplish the exploratory search tasks.
- They were less confident about the fact they fulfilled the exploratory search

tasks.

- The exploratory search task completion supposed to identify multiple items, contrary to the structured ones.
- They changed far more often what they were looking for during the exploratory search tasks.

It proves that even if it is difficult it is possible to create search tasks that are exploratory to a certain extent. In [195] the authors identified several attributes required for an exploratory search task, based on a review of existing literature at the time of writing (2012):

- The work tasks that are the focus of the simulations should be oriented toward learning and investigation tasks. They may include everyday life information problems; a work task does not have to be completed within a "work" setting.
- The context and situation for the work task should be clearly specified; the topic or indicative request is an opportunity for introducing some ambiguity. Topic assignments that are open-ended and/or target multiple items as results are more likely to elicit exploratory search behaviors. A balance needs to be struck between the standardization required for an experiment (in which each subject is performing the same assigned task) and the inherent flexibility of exploratory search.
- Multiple facets should be included in the simulated work task situations and the search topics. Introducing multi-faceted search tasks will serve the dual purpose of making the simulated work task situations more realistic and ensuring that they are not too simple to evoke exploratory search behaviors.
- Possibilities for eliciting dynamic multi-stage searches should be considered. The most obvious approach would be to write the simulated work task situations as involving multiple stages; however, this approach will not capture the types of changes in the search processes that might be invoked by changes in the searcher's understanding of the problem. Longitudinal study designs would be useful, even if difficult to implement.
- Data collection and evaluation methods should be attuned to the goals and attributes of exploratory search tasks. Particularly for studies related to system design, the resulting system will be more effective if it can provide seamless support through searching and into information organization, analysis, and sense-making use.

2.4 Conclusion

Even if search engines are very popular and intensively used there is still an important room of improvement concerning complex search query and need, includ-

ing exploratory search ones. The actual major search systems are optimized for lookup tasks and do not offer a sufficient support for complex information needs. Consequently the users rely today on their own search strategies to perform exploratory search. The strategies of compensation they employ are cognitive consuming. An interesting research dynamic started around human-computer information retrieval systems and exploratory search in particular. It is important to improve the performance of search engines for these tasks as it is notably related to learning and decision-making. Today there is still a lot of research on the exploratory search definition itself. It is difficult to characterize it as the high user involvement in the search process makes the classic information retrieval models unadapted. There is consequently an important heterogeneity in the systems proposed today. The evaluation of these systems is also a hard point for many reasons. It is difficult to design task-based scenario for exploratory search. The results among the users are difficult to compare as they are very subjective. The applications themselves are also difficult to compare as they propose very diverse search experience and design. Exploratory search engines evaluation is one of the most opened questions in the field. Meanwhile interesting exploration applications and functionalities are popular. It includes the search engines knowledge panels, the Facebook Graph Search and Stumble Upon. They familiarize the users with such search approaches and open the door to more innovative solutions. However, many research questions remain opened and some of them are addressed in this thesis. The exploratory search needs are complex to solve and can greatly benefit from the most advanced search techniques. The incorporation of structured semantics in search, referred to as *semantic search*, is one of the most promising approach to solve complex information needs by enhancing the algorithms and users' interactions. Semantic search is detailed in the following chapter.

CHAPTER 3

Semantic search

Contents

3.1 Introduction

The search applications are dependent on the web content quality. Complex information needs such as exploratory search ones can be solved more efficiently by processing machine-understandable content. An important trend impacting the search engines today is the proliferation of structured data on the web. These data are published using microformat, microdata or semantic web formats. The machine readability of structured data is a central motivation for the semantic web vision: *"an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation"* [15]. In this context, the Linked Open Data ongoing initiative is particularly of interest. It leads to massive data publications in semantic web formats, constituting the Linked Open Data cloud. Recently end-users applications using linked data sources as background knowledge appeared. They demonstrated the maturity of such data, especially DBpedia, as a valuable knowledge source for end-user purposes.

Search is deeply impacted by the increasing amount of publicly accessible structured data on the web. The incorporation of semantics in search is referred to as *semantic search*, which is a very active research topic. Meanwhile semantic

searches functionalities and applications are deployed over the web, notably by major players. Consequently the users are more and more familiar with it. Semantic search can take many forms. It can target a wide range of users' information needs and is particularly promising for solving complex information needs including exploratory search. The semantics can be used to enhance various component of a search system including its processing capabilities and interface.

In this chapter we will review (3a) the importance of the structured data publication trend today, (3b) the semantic web vision and realizations including the Linked Open Data initiative, (3c) the existing semantic search approaches and their actual deployment through industrial initiatives.

3.2 Structured data proliferation

Today, more and more structured data are embedded in HTML pages. An increasing amount of websites use markup standards to identify various objects in their content such as dates, events, reviews or people. For this purpose websites developers employ markup standards such as Microformat[1], Microdata[2] and RDFa[3]. The **microformats** were the first semantic markup in HTML that were widely used. It is a grass-root effort driven by an online community. Microformats are not standards. **Microdata** is standardized by the W3C and allows to embed semantic content in web pages. **RDFa** (RDF in attributes) is a W3C recommendation that enables to embed RDF triples in HTML pages. For an extensive presentation of RDF, the reader may refer to the *Semantic web* section of the chapter.

The publication of structured data is encouraged by major web players such as Bing[4], Facebook[5], Google, Yahoo[6] that process and make visible these data within their applications. In 2008 Yahoo launched SearchMonkey[7], an open platform that allowed web developers to enhance the appearance of their websites in the Yahoo search results. The services used, among others, the Microdata and RDFa formats. Embedded metadata were used to enrich the search results presentation. The service was shut down in 2010. In 2012 Bing, Google and Yahoo and Yandex[8] released schema.org[9]. Schema.org proposes a microdata vocabulary supported by these major search engines in order to offer more structured results to the users. In the following source code extract the schema.org classes *Movie*[10], *AggregateR-*

[1] http://microformats.org/
[2] http://www.w3.org/TR/microdata/
[3] http://www.w3.org/TR/rdfa-core/
[4] http://www.bing.com/
[5] http://www.facebook.com/
[6] http://yahoo.com/
[7] http://developer.yahoo.com/searchmonkey/siteowner.html
[8] www.yandex.com
[9] http://schema.org/
[10] http://schema.org/Movie

ating[11], *Person*[12] and related properties are visible. These information are used in search engine rich snippets to enhance the results presentation, see figure 3.1. In 2014 schema.org evolved to allow the websites to describe the actions they enable and how these actions can be invoked[13] e.g. make a reservation in a restaurant.

Listing 3.1: Extract of schema.org microdatas present in IMDB's 2001 : a space odyssey HTML page

```
<div id="pagecontent" itemscope itemtype="http://schema.org/Movie">
...
<div class="star-box-details"
itemtype="http://schema.org/AggregateRating"
itemscope itemprop="aggregateRating">
            Ratings:
<strong><span itemprop="ratingValue">8,3</span></strong>
from <a href="ratings?ref=tt_ov_rt"
title="293 491 IMDb users have given a weighted average vote of 8,3/10" >
<span itemprop="ratingCount">293 491</span> users
</a> 
</div>
...
<div class="txt-block" itemprop="director"
itemscope itemtype="http://schema.org/Person">
    <h4 class="inline">Director:</h4>
        <a href="/name/nm0000040/?ref_=tt_ov_dr" itemprop='url'>
        <span class="itemprop" itemprop="name">Stanley Kubrick</span></a>
</div>
...
```

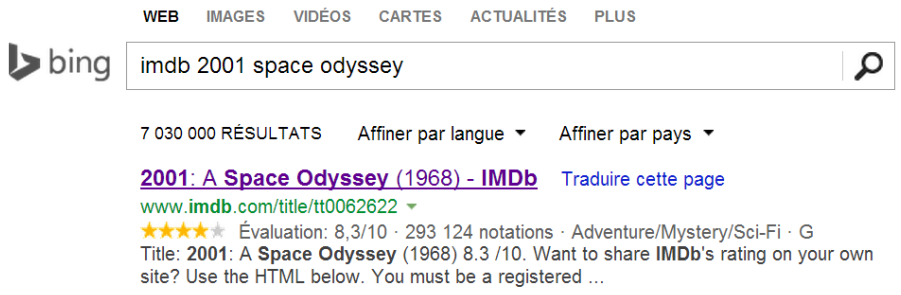

Figure 3.1: Example of schema.org microdata use on the Bing results page

Meanwhile Facebook played an important role for the adoption of RDFa. Their Open Graph Protocol (OGP)[14] is based on RDFa. The social network released the OGP in 2010 when they started the distribution of the *like* button all over the web. Third-party websites have the possibility to integrate the like button on their content in order to be visible inside Facebook and benefit from its social activity.

[11] http://schema.org/AggregateRating
[12] http://schema.org/Person
[13] http://blog.schema.org/2014/04/announcing-schemaorg-actions.html
[14] http://ogp.me/

Thanks to the RDFa markups contained in the pages, the social network is able to correctly integrate the third-party content by recognizing its nature (e.g. a film) and its components (e.g. title, description, depiction). In addition to schema.org microdata IMDB also uses the OGP vocabulary in its HTML pages, see figure 3.2.

Listing 3.2: Extract of OGP RDFa present in IMDB's 2001 : a space odyssey HTML page

```
1  xmlns:og="http://ogp.me/ns#"
2  ...
3  <meta property='og:type' content="video.movie" />
4  <meta property='fb:app_id' content='115109575169727' />
5  <meta property='og:title' content="2001_:_a_space_odyssey_(1968)" />
6  <meta property='og:site_name' content='IMDb' />
7  ...
8  <meta property="og:description" content="Directed␣by␣Stanley␣Kubrick...
```

Figure 3.2: Example of third-party content integration in Facebook thanks to RDFa through OGP

[17] performed an analysis over 3 billion web pages originating from 40 million websites. They found that 50% of the top 10.000 websites, according to Alexa analytics[15] embed structured data. They also discovered structured data within 369 million out of the 3 billion pages contained in the corpus (12.3%). These pages originated from 2.29 million websites among the 40.6 million websites, identified by their pay-level-domain (5.64%). The study also confirmed that the vocabularies proposed by the major players are very popular: OGP for RDFa and schema.org for Microdata. An important traction comes from the visibility of structured data through popular applications, forming a virtuous circle.

[15] http://www.alexa.com/

3.3 Semantic web

Tim Berners-Lee, the web inventor, is the main semantic web initiator. He publicly highlighted the need for semantics in the web at WWW conference plenary in 1994[16]. He proposed a high-level plan of its architecture in 1998[17]. The semantic web vision became more visible thanks to an article published in 2001 in the Scientific American [15]. This article attracted the attention of many researchers and triggered a community-wide effort to realize this vision. Semantic web can be defined as the web augmentation by formal metadata giving to software the access to some semantic facets of information. Another definition is *the semantic Web is about exposing structured information on the web in a way that its semantics are grounded on well-defined and agreed-upon vocabularies* [101]. In others words it aims to increase and enhance the web functionalities by enabling readability of its information. Tim Berners-Lee put forward the idea to use the web architecture and principles to express the meaning contained in the web pages. By exploiting the resulting formal metadata the software agents reach a level of understanding that makes possible a new level in process automation and information access. The semantic data introduced are complementary to the original graph of documents and does replace it.

In 2001, the web was mostly a huge graph of documents i.e. of web pages. When the semantic web vision was proposed the question of how to represent the semantic information on the web arose. The reflection and elaboration of the models and formalisms began. Tim Berners-Lee proposed the semantic web layer cake at the XML 2000 conference, see figure 3.3. It can be considered as the semantic web technological roadmap. All the layers are standards and are developed within the W3C. The semantic web is extending the document layer of the web by a data layer, expressed in RDF, and an ontology layer. They are described hereafter.

The first layer is the **Uniform Resource Identifier / International Resource Identifier**[18] one. This standard allows giving a unique identification of a resource on the web.

The **Resource Description Framework** is a graph model to describe resources. It was initially expressed in XML syntax but later other syntaxes were proposed[19] e.g. Turtle and RDFa. RDF is the bedrock of the semantic web. It is used to express simple statements about resources. RDF is a basic assertion model allowing the expression of triples in the form *subject - predicate - object* e.g. *Claude Monet - isBornIn - Paris*. The subject is always a resource identified by its URI. The object is a resource or a primitive value. RDF is simple and flexible and constitutes a solid basis for more expressive languages.

The SPARQL language [155], recursive acronym for SPARQL Protocol and RDF Query Language, allows to query an RDF base through its SPARQL endpoint.

[16] http://www.w3.org/Talks/WWW94Tim/
[17] http://www.w3.org/DesignIssues/Semantic.html
[18] http://www.w3.org/Addressing/
[19] http://www.w3.org/TR/rdf-syntax-grammar/

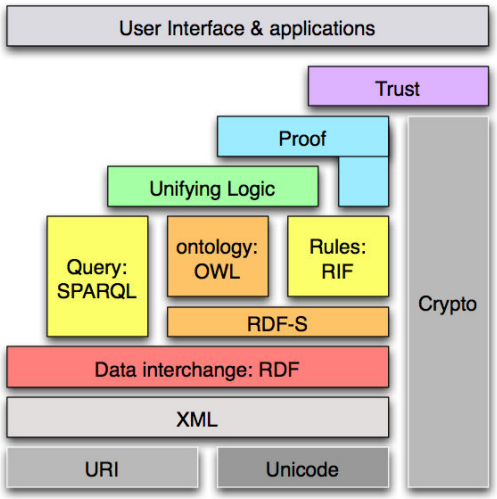

Figure 3.3: The semantic web layer cake

The syntax of the language is inspired by the Turtle one and is close to SQL one. SPARQL is a key technology for the semantic web; it is an official W3C recommendation since 2008[20]. SPARQL queries are in the form of graph patterns that include variable(s) instead of subject(s) and/or predicate(s) and/or object(s). The SELECT clause specify the variable(s) to retrieve and the WHERE clause receive the graph pattern to match with the graph queried. SPARQL is very expressive and allows the expression of unions, intersection, optional clauses, regular expression filtering, basic mathematic computation, grouping, sorting. It also allows to retrieve graphs instead of results sets thanks to the CONSTRUCT clause.

There is a need for more descriptive levels to support advanced reasoning. This is the purpose of the ontological layer. An ontology is a partial representation of a world's conceptualization [64] or in other words the conceptual vocabulary of a domain. It is a necessary representation to exchange the data between the applications and to support automatic reasoning by giving descriptions on the resources themselves. Ontologies support shared understanding of domains of interest [185]. They avoid the conceptual and terminological confusions among the users and software agents. Several semantic web ontology models exist; they differentiate themselves in their level of expressivity. **RDF-Schema** (RDFS)[21] uses the RDF specifications and extends it in order to support basic reasoning. Its expressivity is limited but powerful. RDF notably allows the expression of classes and properties hierarchy (thanks to *rdfs:subClassOf* and *rdfs:subPropertyOf*). It allows specifying the properties domain (the class of resources that the property should be used to describe) and range (the class of resources that should be used as values for that property). The second semantic web ontology language is the **Web**

[20]http://www.w3.org/TR/rdf-sparql-query/
[21]http://www.w3.org/TR/rdf-schema/

Ontology Language[22] (OWL). OWL is has several levels of complexity: EL, QL and RL[23]. As the expressivity of the model raises the automation of the reasoning becomes more powerful but also more complex. A version 2.0 of OWL is currently being developed[24]. These ontology languages allow to make inferences on RDF data i.e. producing new knowledge from existing knowledge.

Today, the top levels of the semantic web architecture (trust and crypto) are not stable and subject to active research.

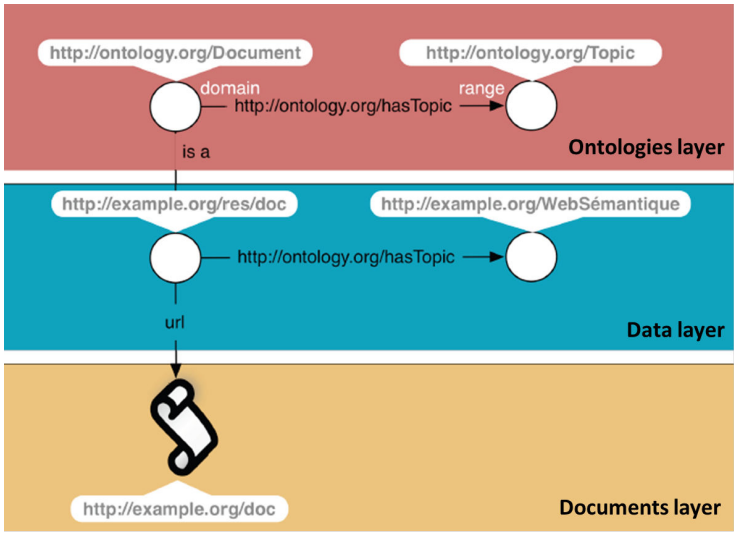

Figure 3.4: The documents layer is completed by data and ontologies ones, taken from [173]

3.4 Linked data

3.4.1 Principles

Once the semantic web concepts and models were sufficiently mature the research community started to think about the ways to publish a large amount of RDF data. There was a need to demonstrate the vision and gains traction within the developer community. Initiatives were launched to design mapping languages having the objective to convert relational data into linked data e.g. the W3C RDB2RDF[25]. At this time the publication of semantic data was difficult and required expensive translation processes. It resulted in moderate growth, sparsity and poor demonstration of value capabilities. The most successful initiative in terms of semantic web data publication started in 2007: the Linked Open Data project. The linked open data project is a community initiative launched by the W3C that aims to

[22] http://www.w3.org/TR/owl-features/
[23] http://www.w3.org/TR/owl2-profiles/$OWL_{2E}L$
[24] http://www.w3.org/TR/owl2-overview/
[25] http://www.w3.org/TR/r2rml/

identify datasets under open licenses and to republish it relying on the principles of the web and using the semantic web formalisms. The conversion is performed in a grass-root manner: data first, schema second (often derived from data) and logic third. The project also aims to connect these datasets among themselves. By releasing massive and useful linked data sources, this initiative gave a fresh boost to the semantic web research community and increased its public visibility. It helped solving the *chicken-and-egg* problem encountered by the community: the lack of data leads to the lack of applications which leads of the lack of data, etc. To be in line with this success the W3C semantic web activity has been replaced by data web activity in 2013[26]. Tim Berners-Lee introduced in 2006 four principles to follow for the their publication[27]:

- *Use URIs to denote things.*
- *Use HTTP URIs so that these things can be referred to and looked up ("dereferenced") by people and user agents.*
- *Provide useful information about the thing when its URI is dereferenced, leveraging standards such as RDF*, SPARQL.*
- *Include links to other related things (using their URIs) when publishing data on the Web.*

Four years later, 5 complementary principles were proposed. They are less technical and are in the sense of the open data movement which aims to release public data for the good of societies (data about education or public expenses for instance):

- *Available on the web (whatever format) but with an open license, to be Open Data.*
- *Available as machine-readable structured data (e.g. excel instead of image scan of a table).*
- *as (2) plus non-proprietary format (e.g. CSV instead of excel).*
- *All the above plus, Use open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff.*
- *All the above, plus: Link your data to other people's data to provide context.*

Microformats and microdata publish the descriptions of entities that are distributed all over the web and isolated; on the contrary linked data states explicitly the relationships between the entities. Data are described through RDF links that go through the servers and connect all the data in a single global graph called the linked open data cloud[28].

[26] http://www.w3.org/blog/data/2013/12/13/welcome/
[27] http://www.w3.org/DesignIssues/LinkedData.html
[28] http://linkeddata.org

3.4.2 Schemas

It is a common and good practice to reuse existing third-party ontological vocabularies when publishing new data. It increases the homogeneity and interoperability of the distributed data sources. The data publishers consequently combine terms from existing vocabularies with proprietary ones covering their specific needs. The Linked Open Vocabulary project shows the dependencies between the vocabularies[29].The vast majority of the linked data sources have ontology vocabularies that uses terms from OWL and RDFS. Some others schemas, described hereafter and used in following works of this thesis, are also well represented in the linked open data cloud:

The **Dublin Core Metadata Initiative**[30] provides a set of metadata for describing numeric and physical resources. Widespread properties in the linked open data cloud include: *dcterms:creator* to specify the maker of the resource (e.g. a person, a company), *dcterms:date* to specify a date related to the resource, *dcterms:description* to add a description of the resource and *dcterms:subject* that specify the topic of the resource in the form of a text, a code or better another resource.

The **Friend-Of-A-Friend** project (FOAF)[31], started in 2010, proposes a widespread vocabulary that serves to describe persons, their social links and relations to others objects (e.g. authoring). Several social networks use the FOAF vocabulary to expose their data: LiveJournal[32] (social network and blogging) and Identi.ca[33] (open source micro-blogging service) for instance. The FOAF data distributed over the web constitutes a decentralized social network following the distributed principles of the Web. The FOAF profiles are regularly crawled in order to support researches.

Semantically-Interlinked Online Communities (SIOC)[34] is together with FOAF the most important vocabulary when considering the socio-semantic web. SIOC was released in 2004 and aims to link the communities content on the web. It provides a vocabulary for describing common resources in platforms such as forums, blogs or mailing list (e.g. user accounts, messages, threads). SIOC aims to break the *"walled garden"* of the social networks by proposing an open vocabulary increasing the interoperability between the platforms [23]. In fact today the users are isolated within numerous platforms (one account per application) that are independent from each other's. SIOC exporters for Wordpress[35], Drupal[36] and vBulletin[37] are available.

[29]http://lov.okfn.org/dataset/lov/search/s=dbpediaof=30
[30]http://purl.org/dc/terms
[31]http://www.foaf-project.org/
[32]http://www.livejournal.com/
[33]https://identi.ca/
[34]http://sioc-project.org/ontology
[35]http://wordpress.com/
[36]https://drupal.org/
[37]https://www.vbulletin.com/

Simple Knowledge Organisation System (SKOS)[38] is an RDFS schema used to describe concepts hierarchies such as taxonomies, folksonomies or thesauri. The concepts are linked by properties representing a subsumption hierarchy. These properties include *skos:broader* and its opposite *skos:narrower*. They are powerful to describe topic ontologies where the semantic of *rdfs:subClassOf* is inappropriate. Indeed the subclass relationships can not be applied for topic taxonomies. For instance *Pointillist_Painter* is not a valid subclass of *Impressionist_Painter*. The property *skos:broader* is appropriate in this case. SKOS also proposes the more general *skos:related* property that allows declaring that an association exists between two concepts.

Basic Geo Vocabulary[39] is a simple RDFS vocabulary that notably allows the representation of latitude (*geo:lat* property) and longitude (*geo:long* property) following the World Geodetic System 84 standard[40].

3.4.3 Datasets

The amount of linked data triples uploaded on the web is actually constantly growing. They originate from various initiatives driven by research, private and governmental actors. There is an important representation of the domains having a strong experience in knowledge representation and classification. The life sciences data and libraries ones are well represented [19]. In 2014 the graph of the linked data has an estimated size of 100 billion triples [101]. The governments are increasingly attentive to this technological shift. There is a growing amount of open data initiatives. The United Kingdom platform[41] makes use of the semantic web technologies.

A reflexion around linked data economy and market start to emerge. [11] stresses the absence and the need of a lively economic environment for linked data. In the opinion of the author it is time to transform the actual enormous amount of data obtained from research into a virtuous circle of value creation. [207] goes further by proposing a market-based SPARQL broker that identify the best dataset to query (free or paying ones) according to quality and cost inputs. The main finding of their simulations is that a mixture of free and commercial providers results in the best market performance for both data users and producers. Linked data is also an important topic of research for the Web Science community[42]. Web Science is a multidisciplinary community (computing, physical, social sciences) that *studies the vast information network of people, communities, organizations, applications, and policies that shape and are shaped by the Web, the largest artifact constructed by humans in history*.

Linked data is constantly evolving material: new sources emerge online and

[38] http://www.w3.org/2004/02/skos/
[39] http://www.w3.org/2003/01/geo/
[40] http://en.wikipedia.org/wiki/World_Geodetic_System
[41] http://data.gov.uk
[42] http://websci14.org/exhibit

the datasets are updated as well as their connections. The fact to connect several knowledge sources that were previously isolated allows performing queries taking advantage of such connections that were hard or impossible before. The semantic datasets are mainly available in the form of raw files or through the SPARQL endpoint of triple-stores. Triples are graph-oriented database which are close to tradition relational database management system in term of functionalities.

A central and emblematic initiative in the linked open data cloud is DBpedia. DBpedia is the RDF semantic conversion of Wikipedia. The idea of using Wikipedia to produce linked data was initially proposed by [186] with an extension of the MediaWiki software allowing the contributors to specify semantic relations while writing the content. It is a collaborative approach. The DBpedia project has a totally different approach as it performs an automated extraction. It was started by the Free University of Berlin, the University of Leipzig and the OpenLink Software company in 2007. The authors observed that there were at that time only punctual initiatives limited to defined domains with closed schemas [6]. They stressed the need to create a global traction by proposing a cross-domain data source, useful and accessible to a general audience as well as to make a solid training ground available to the research community. Before DBpedia the researches were restricted to manually crafted and domain-specific datasets. As it is difficult to design a generic schema for a web-scale cross-domain data source they propose a grass-root approach: they extracted and deduced the ontology from the data. They applied this method to Wikipedia data to build DBpedia. Wikipedia offers several decisive advantages:

- The data are available under a free license (Creative Commons[43]).
- Due to its encyclopedic nature it covers numerous topics and a wide range of domains. At the time of the writing there are more than 4 million articles in the English version.
- It is a negotiated, consensual knowledge thanks to its collaborative writing process. It represents a community agreement to a certain extent.
- It is continuously updated as it is written by a large and active community of contributors.
- It is mainly constituted of text but over the years the collaborators also describe many structured data that constitute excellent candidates for an RDF conversion.

DBpedia can be considered as the machine-understandable equivalent of Wikipedia. It makes this collection of encyclopedic knowledge, previously only accessible for human reading, usable by machines. The couple Wikipedia-DBpedia is unique on the web. It offers both the same information on textual and structured forms about a wide range of topics. Moreover this knowledge is accessible

[43]http://en.wikipedia.org/wiki/Wikipedia:Text_of_Creative_Commons_Attribution-ShareAlike_3.0_Unported_License

in several languages. Thanks to its very wide coverage DBpedia is also the key interlinking hub of the Linked Open Data cloud. The new datasets link their data to DBpedia in order to be more visible and to be indirectly connected to others sources.

Technically the conversion of Wikipedia pages to RDF is performed thanks to a set of extractors. At the time of writing there are sixteen extractors[44] targeting various part of the HTML pages: e.g. title, image, geographic coordinates. An important extractor is the infobox one that identifies and converts the data contained in Wikipedia infoboxes[45]. The infoboxes are the set of structured data displayed on the top-left of Wikipedia pages, see figure 3.5. They constitute information patterns used on the pages for a particular topic class (e.g. the painters). The DBpedia community creates mappings[46] that specify the correspondences between the infoboxes properties and the DBpedia ontology: e.g. the *born* field corresponds to the *http://dbpedia.org/ontology/birthDate* and *http://dbpedia.org/ontology/birthPlace* properties. The extractor identifies in this case the bith place and data that are mixed in a single text field in Wikipedia. The conversion results in a set of triples having the page's topic as object, see figure 3.6. The extraction is run periodically. As the number of mappings raises the DBpedia ontology becomes more detailed and the data quality increases. A *live* version[47] of DBpeda publishes RDF sets that list the changes within Wikipedia minutes by minutes. The online encyclopedia in continuously updated and some applications require the freshest data, for instance applications based on the news. DBpedia 3.9 was released in september 2013 and it offers[48]:

- 529 classes (DBpedia 3.8: 359)
- 927 object properties (DBpedia 3.8: 800)
- 1290 datatype properties (DBpedia 3.8: 859)
- 4 million things described including 832.000 persons, 639.000 places, 372.000 creative works (including 116.000 music albums, 78.000 films and 18.500 video games), 209.000 organizations (including 49.000 companies and 45.000 educational institutions), 226.000 species and 5.600 diseases
- A total of 470 million triples

An important part of the DBpedia knowledge is captured by the hierarchy of categories. In Wikipedia the articles are classified into categories that appear at the bottom of the pages in order to assist browsing. For instance Claude Monet[49] belongs to, among others: *artists from Paris, french impressionists painters, alumni of the école des beaux-arts*. The articles are usually placed in the most specific categories

[44] http://wiki.dbpedia.org/DeveloperDocumentation/Extractor
[45] http://en.wikipedia.org/wiki/Category:Infobox_templates
[46] http://mappings.dbpedia.org/index.php/Mapping_en
[47] http://live.dbpedia.org/
[48] http://blog.dbpedia.org/
[49] http://en.wikipedia.org/wiki/Claude_Monet

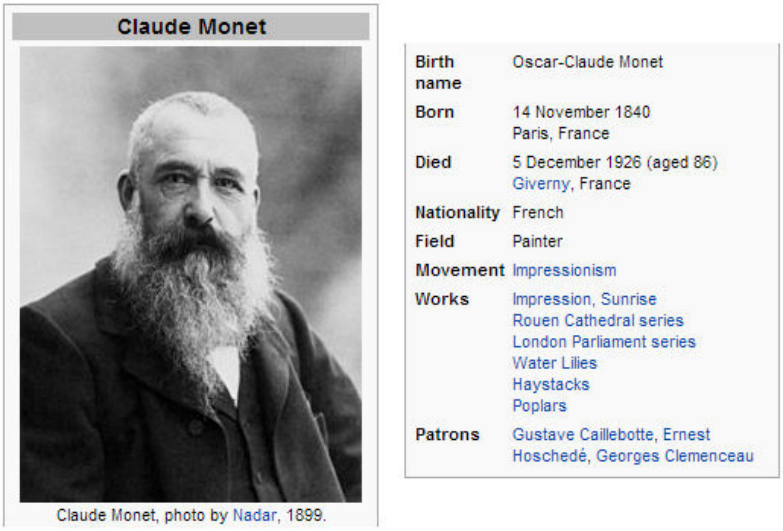

Figure 3.5: Information contained in the Claude Monet Wikipedia's page infobox

dbpprop:birthName	Oscar-Claude Monet
dbpprop:birthPlace	dbpedia:Paris
dbpprop:caption	Claude Monet, photo by Nadar, 1899.
dbpprop:dateOfBirth	1840-11-14 (xsd:date)
dbpprop:dateOfDeath	1926-12-05 (xsd:date)
dbpprop:deathDate	1926-12-05 (xsd:date)
dbpprop:deathPlace	Giverny, France
dbpprop:field	Painter
dbpprop:hasPhotoCollection	http://wifo5-03.informatik.uni-mannheim.de/flickrwrappr/photos/Claude
dbpprop:imagesize	200 (xsd:integer)
dbpprop:influencedBy	dbpedia:Johan_Jongkind dbpedia:Eugène_Boudin dbpedia:Gustave_Courbet
dbpprop:movement	dbpedia:Impressionism
dbpprop:name	Claude Monet Monet, Claude Oscar
dbpprop:nationality	French
dbpprop:patrons	dbpedia:Gustave_Caillebotte dbpedia:Georges_Clemenceau dbpedia:Ernest_Hoschedé
dbpprop:placeOfBirth	Paris, France

Figure 3.6: Extract of DBpedia triples having the Claude Monet DBpedia resource as object

to avoid an explosion of their number. The DBpedia categories can be considered as an orthogonal knowledge structure in DBpedia. They are typed as *skos:concept* and linked each-others thanks to the *skos:broader* property. Several DBpedia-based applications make use of the categories hierarchy to make sense out of the data. An illustration of the categories hierarchy, as well as the DBpedia use of third-party vocabularies can be seen on figure 3.7.

Wikipedia is an international project, written in 287 languages[50]. The DBpedia resources' labels and abstracts (short descriptions) are stored in multiple languages (120). It is consequently easy for applications that use DBpedia as a

[50]http://meta.wikimedia.org/wiki/List_of_Wikipedias

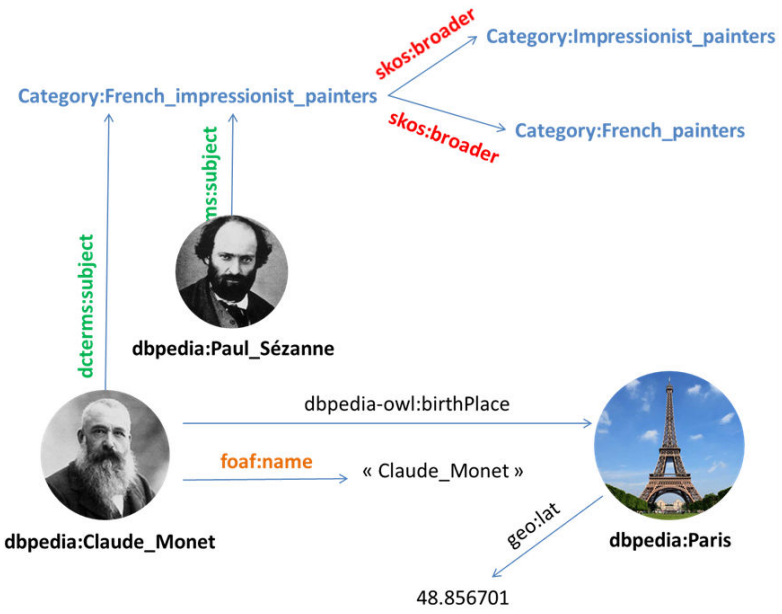

Figure 3.7: Illustration of the categories hierarchy and use of third-party vocabulary in DBpedia

knowledge source to switch from one language to another. Moreover Thanks to local initiatives 15 international language-specific DBpedia versions are extracted from corresponding Wikipedias and accessible through SPARQL endpoints[51]. As the languages are related to cultures, these local chapters significantly vary in the knowledge they capture[52]. It is consequently important that this movement continues in order to offer the best knowledge coverage in the international context of the web. For instance it is more interesting to query the French-speaking DBpedia SPARQL endpoint[53] than English-speaking one when considering the French museums, see figure 3.8.

Several tools help the creation of semantic content and applications. As mentioned before the semantic annotations volume significantly increases on the web. But at the time a vast amount of web pages are not semantically annotated. Several commercial and research tools allow identifying semantic resources contained in raw textual document such as web pages. This process is known as *named entity recognition*. It is useful as it creates semantic views of unstructured documents. Then this extra-knowledge can be used to enable processing that requires a semantic description of the content. Popular solutions include OpenCalais[55], Alchemy[56],

[51]http://wiki.dbpedia.org/Internationalization/Chapters/
[52]http://wiki.dbpedia.org/Datasets39/CrossLanguageOverlapStatistics
[53]http://fr.dbpedia.org/sparql
[55]http://www.opencalais.com/
[56]http://www.alchemyapi.com/

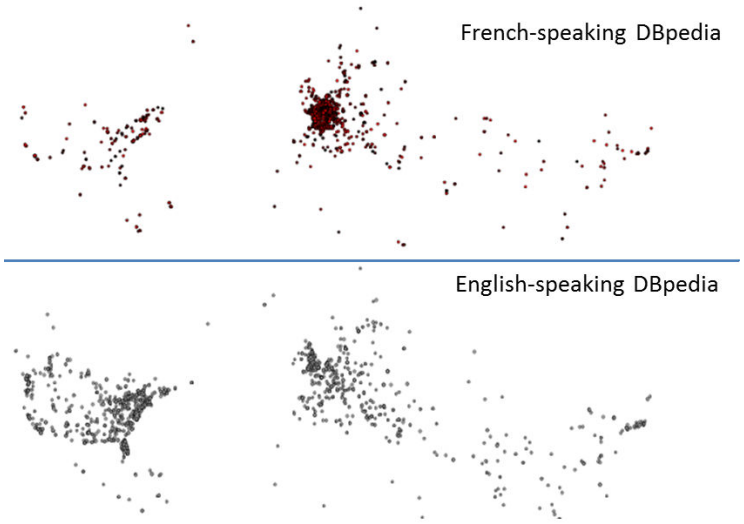

Figure 3.8: Comparison of geo-tagged museums location in French and English-speaking DBpedia versions[54]

Zemanta[57], DBpedia Spotlight [58] and Stanbol[59]. Some of these systems operate on DBpedia. DBpedia constitutes a valuable knowledge to solve the natural language polysemy by using contextual elements [172] e.g. recognizing that a web pages deals with the Apple company and not with the fruit. The uniqueness of the URI used in the web of data eases the disambiguation. Several works mentioned in this thesis make use of named entity recognition. Another popular tool based on DBpedia is the lookup[60]. It enables auto-completion composed of DBpedia resources starting from a string, when typing a query for instance. In other words it is a rapid keyword resource search/selection.

The DBpedia automated conversion process is powerful and results in a large and useful dataset. Nevertheless it also leads to inconsistencies and errors in data. Basically wrong information present in Wikipedia will lead to wrong data in DBpedia. The Wikipedia collaborators also make errors or approximations in the writing process itself or in information structuring that will be crystallized in RDF. It is mainly due to misuse of Wikipedia infoboxes [190], another known problem is the presence of cycles in the hierarchy of categories e.g. *morals* placed under *ethics* and inversely [177]. The errors and inconsistencies are an important problem for the linked data initiative which is mainly based on automated conversions of large public data sources. Research works that detect and correct these inconsistencies are essential, e.g. [190]. It is especially critical for end-user applications where wrong knowledge can be exposed and lead to a loss in confidence by the users. Others works aim to improve the dataset by inferring missing information that are

[57]http://www.zemanta.com/api/
[58]http://spotlight.dbpedia.org/
[59]http://incubator.apache.org/stanbol/
[60]http://wiki.dbpedia.org/Lookup

not present in the original data source e.g. automatic typing of resources [56]. [179] proposes to use DBpedia data to improve Wikipedia, creating a virtuous circle.

3.4.4 Applications

The applications that use linked data as background knowledge has to face two main problems today:

- Semantic heterogeneity: a difficult point is that different data sources use different vocabularies to represent similar data. For instance, at a time the DBpedia Spanish-speaking version[61] used *foaf:name* instead of *rdfs:label* to store the resources labels (now corrected). Another problem is that a vocabulary can be used in different ways by several data sources e.g. associating different types of values to an identical property. The LOD is a community effort and it is difficult to prevent such inconsistencies at web-scale.
- Data quality: the web of data, like the web, contains outdated, conflicting and as mentioned before wrong data. The W3C has launched the Provenance Working Group[62] in order to propose solutions to track the provenance of the data. Mechanisms to assess the data quality and truth-worthiness are increasingly needed. The Prov-Ontology became a W3C recommendation in April 2013[63].

Dadzie identifies the following challenges for linked data use [41]:

- **Exploration starting point**: where to start; existing LD browsers assume the end user will start browsing from a specific, valid URI. How can a vizualisation starting point be presented to users in such a way that it is meaningful?
- **Combating information overload**: presenting end users with all the properties of a given resource, along with the relations through which the resource is linked to other entities, leads to information saturation and a dense information space. How can we present this information in a more legible form?
- **Returning something useful**: RDF is the staple recipe for resource descriptions, returning information using this knowledge representation format inhibits comprehension. How can RDF, and the information contained within + result object instance descriptions, be represented in a more legible, manageable form?
- **Enabling interaction**: end users are familiar with the makeup of the Web and its browsable nature. Is it possible to replicate such familiarity which users experience when browsing the WWW on the WoD (Web of Data)?

[61] http://es.dbpedia.org/sparql
[62] http://www.w3.org/2011/prov/wiki/Main_Page
[63] http://www.w3.org/TR/prov-o/

Several end-user applications make use of linked data datasets and DBpedia in particular to enable new user's experiences. These realizations include Seevl[64] a music discovery platform displayed on figure 4.30, the BBC website augmentation by linked data[65] and *Everything is connected*[66] that creates a short narrative film explaining how the user is indirectly connected to an item of his choice, starting from his Facebook profile.

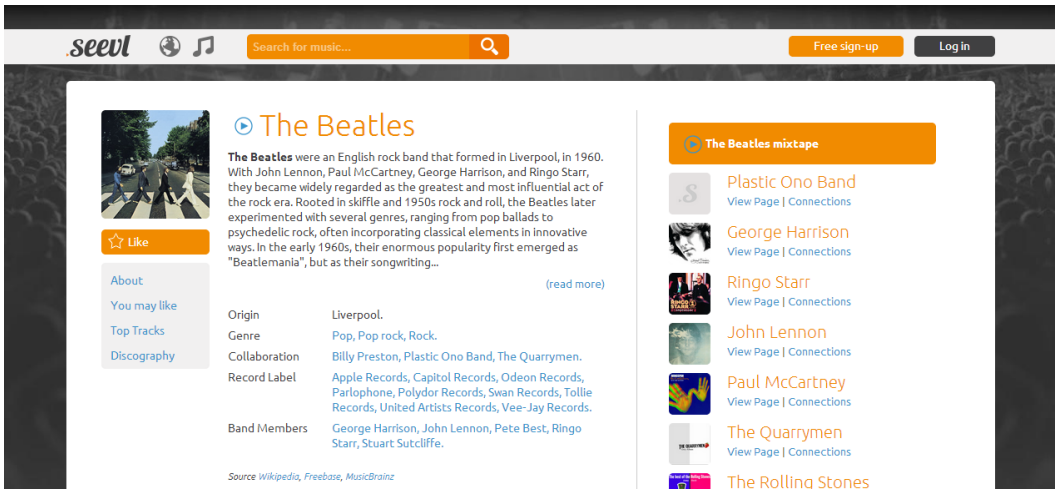

Figure 3.9: The Beatles page in Seevl, a DBpedia-powered discovery platform

3.5 Search with semantics

For a long time web search algorithms have been based on textual similarity and web pages graph structural criteria. The use of refinements strategies such as personalization or collaborative filtering help to retrieve better results but there is still an important room for improvement. As stated in [8]: *"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"*. The semantic data sources allows to elaborate better representation and modeling of the human cognition as well as of the content. One definition of semantic search is *a retrieval paradigm that first, makes use of the structure of the data or explicit schemas to understand user intent and the meaning of content and; second exploits this understanding at some part of the search process*[67]. Major search engines have made heavy investments on computational power to tackle scalability and speed of service issues. It does not help to satisfy complex information needs. Complex search tasks require a better understanding of the user's intent and information/data collection that is searched. Using the seman-

[64] http://seevl.fm/
[65] http://www.w3.org/2001/sw/sweo/public/UseCases/BBC/
[66] http://everythingisconnected.be/
[67] http://slideshare.net/pmika/making-things-findable

tics helps retrieving relevant results for these tasks and presenting them through intuitive interfaces.

Semantic search (later join by recommendation, see chapter 4) has a major role to play for lowering the technical barriers and help the casual users to benefit from the linked data richness. In fact, until recently the consumption and use of such data was limited to experts that are able to read the RDF syntax and use others technical languages and tools. By the way even technical users face difficulties due to the increasing size and complexity of the linked data datasets. It is more and more difficult to get a good understanding and mental model of them without the support of efficient tools [41]. Increasing the efficiency of semantic search systems is also crucial as the volume of structured data dramatically increases and risks to overwhelm the users and software agents as the web pages did in the mid-nineties i.e. avoid the user to be *lost in the hyperspace* [69]. The availability of a growing amount of semantic data enables new user experiences but at the same time introduces numerous research challenges due to the complexity of integrating them in the search process. Efficient semantic search systems will also demonstrate the large-scale utility of semantic data and encourage their production.

3.5.1 Concepts and approaches

There is an important amount of research works dealing with semantic search, including linked data based approaches. It is a dynamic research area. These works propose solutions to solve the problems and explore the opportunities brought by these rich but complex data. There is an increasing interest to ease the consumption of semantic data by easing search on it, to *hide the semantic web stack* [41] to the users through appealing and intuitive interfaces. The idea of searching thanks to concepts and their meanings rather than literal strings has been investigated by information retrieval research community since the 1980s. Early work that introduced knowledge base to enhance an information retrieval system are mentioned in [36] under the name of *Intelligent Information Retrieval systems*: *"the aim (of these systems) is to use application domain knowledge in the indexing, in the similarity evaluation, or to enrich the query representation"*. These early works date from 1987 [38] and 1992 [67]. Some research propositions are now massively deployed on major player applications, it is the case of [65] that proposed document search augmentation with structured data that is very close to the recent Google Knowledge panel[68], see figure 3.10. The topic is not new and is vast. It consequently gathers the efforts of several research communities including natural language processing, database, information retrieval, knowledge representation and human-computer interaction ones. The difference between traditional search approaches and semantic search is summarized by Koumenides and Shadbolt in [100]:

[68] http://www.google.com/insidesearch/features/search/knowledge.html

Traditional approaches to information retrieval often treat documents as collections or bags of individual words, and their correspondence to a similar representation of user queries generally determines their level of similarity. This notion has often been coupled with simple forms of natural language processing [8] and features based on links, such as popularity and usage, when search is conducted over web-accessible documents. More elaborate retrieval models have also evolved in an effort to include information related to the classification of content inside documents, to prioritize selections based on where query terms are found within the documents (whether part of a title, body, anchor text, etc.). The idea of semantic search is to diverge from this coarse view and sometimes monotonic treatment of documents to a finer perspective, one that is able to exploit and reason intelligently with granular data items, such as people, products, organizations, or locations, whether that is to complement document retrieval or to facilitate different forms of search.

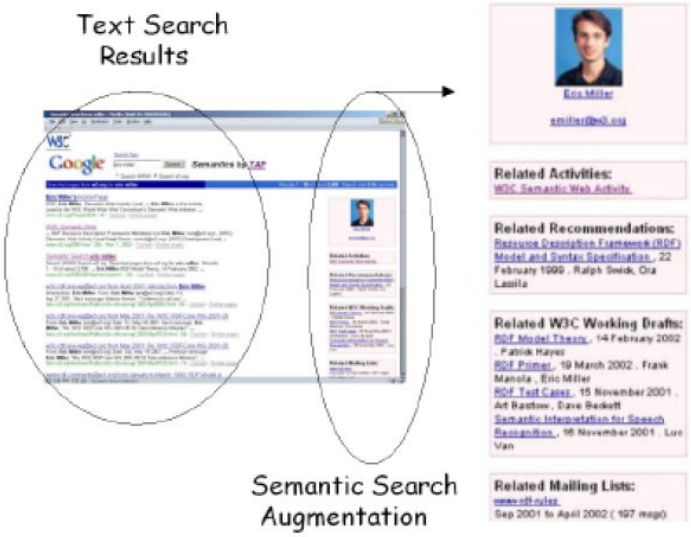

Figure 3.10: Augmentation of search results with semantic web data proposed by Guha and al. in 2003

Numerous semantic search systems were created in industrial and research contexts. [180] and [29] identified the varying aspects of the existing systems. We merged and synthesized them, see figure 3.11, in order to offer an overview on semantic search approaches and to better position our work in the next chapters:

- The **tackled information need** (1): semantic search approaches aim to support a wide range of search tasks and objectives. It is particularly of interest for precise information needs requiring a factual answer that can be retrieved and computed from data. Is is also promising for open search tasks like exploratory search, in this case the meaning of the data is used to focus the

user attention on important information, to structure the interactions and to propose unexpected information paths.

- The **query paradigm** (2) is strongly related to the addressed information need, it includes precise inputs related to an entity or related facts, natural language query and iterative query building. The latter is popular for complex and cognitive-consuming search tasks like exploratory search. In this case the systems rely on schemas and data semantics to set up refinement filters. The query is implicitly or explicitly (query completion, drag and drop) modified/refined along the navigation of the user in the results space.
- The **semantic matching framework** (3): one of the key problematic inherent to semantic search systems is to make an interpretation of the user's information need and to match it with the data collection. The representation of the query is therefore a crucial element of the system. An entity recognition can be performed on the query if is typed in plain text. Some semantic search systems also provide direct entity search (e.g. thanks to a DBpedia lookup). Another possibility is to match the user's query keyword to data collection resource's literals (e.g. labels, descriptions of resources). Several systems also offer natural language querying capacities; the translation of the user's query into the semantic data model is a difficult problem and requires fine exploitation of the semantics. Advanced matching solutions include also semantic adaptations of the vector-space model.
- The **semantic data origin** (4): The data can be directly published/generated in a structured format (e.g. LOD dataset, domain knowledge base) and/or converted from a relational data source and/or obtained thanks to an entity recognition over a set of raw documents. They are often formated in RDF but not exclusively.
- The **semantic model** (5) to understand the data and query can be a knowledge model that represents classes of entities, related attributes and properties (written in OWL or others languages). Lexical models like thesaurus, capturing the semantics at the level of the words are also common.
- The **processing of results** (6): there are a lot of research about ranking algorithms for semantic data. In addition to new algorithm propositions, adaptations of spreading activation [154], TF/IDF (Term-Frequency/Inverse-Document-Frequency), PageRank or HITS (Hyperlink-Induced Topic Search) [95] algorithms are also used in semantic search systems.
- The **retrieved results and interface** (7): the systems can retrieve data, documents or both. Data-retrieval approaches consider ensemble of entities and/or the paths between them as unit of results. The documents/web pages make sense in themselves, when dealing with data it is the data combination, the relations between entities that make sense and that are presented to the user. Thus many semantic search approaches aims to identify in the

mass the most relevant resources that will contextualized the object(s) of interest and create a meaningful, informative ensemble. Depending on the search objective the interfaces can be generic (e.g., tables, trees, lists), fact-specific (e.g., maps, timelines) or entity-specific (e.g. weather in Paris). This shift to a finer granularity is an important research challenge.

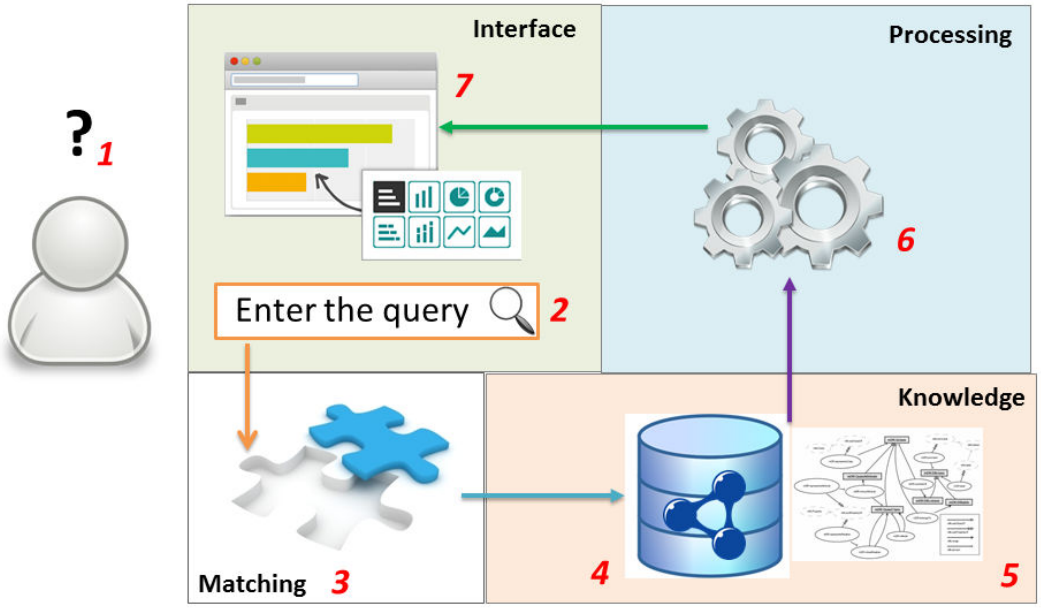

Figure 3.11: Schematic representation of a semantic search engine

3.5.2 Deployment over the web

"Introducing the Knowledge Graph: Things, not strings"[69].

The integration of structured data in the user experience by major players is progressive but now highly visible. The impact on the search process is important and visible to the users. Several applications and functionalities go now further than a simple integration of semantic data through rich snippets. The embellishment of interfaces is a first step that already significantly improve the user experience [66] but semantic search is still at its infancy. In this part we will review important industry initiatives. They demonstrate that search is increasingly moving from words to entities of different types (e.g. persons, places, companies). These systems are proprietary, the algorithms, software solutions and knowledge sources they use are often unknown and publicly inaccessible.

A domain of application of semantic search technologies that have hit the market is the natural language search. This type of search consists of interpreting

[69]http://googleblog.blogspot.fr/2012/05/introducing-knowledge-graph-things-not.html

the user queries that are formulated in natural languages. Natural language processing (NLP) can support tasks like sentiment analysis, question answering and search. The semantics gives a structure that enables a better understanding of the language. Powerset[70], founded in 2006 developed a natural-language search engine based on Freebase and semantic data extracted from Wikipedia[71] through syntactical analysis. It was the first commercial search engine to make use of semantic analysis. The company was acquired by Microsoft in 2008 in order to improve the Bing search engine. **Evi**, previously known as TrueKnowledge, is a company that launched in 2007 an open domain question-answering platform processing a large base combining commonsense, factual, and lexical knowledge to compute the results [182]. An ontology of more than 20.000 classes underlies the knowledge base, which contains typed entites and typed properties. The format used is triple-based and proprietary. The knowledge base was built by importing knowledge from Freebase, Wikipedia (infoboxes and categories), users' collaborations, various databases, and knowledge natural language extraction from raw text. In 2010 it reached 240 million triples about 8 million entities.

Natural language search and question answering functionalities are sometimes coupled with speech recognition to build personal assistant functionalities. Apple's **Siri**[72] was presented to public in October 2011 and is the most popular voice-based personal assistant. Siri interacts with many information and knowledge sources, including WolframAlpha. In January 2012, True Knowledge launched Evi a voice-based mobile assistant based on its NLP technologies for iOS and Android phones. The general public saw an impressive demonstration of natural languages QA with the participation of IBM Watson computer[73] to the American television game show *Jeopardy!*[74]. Watson has natural language capabilities and process knowledge bases to compute its answers [52].

WolframAlpha[75], launched in 2009 by Wolfram research, is a computational knowledge engine that solves users queries by processing hundreds of external and curated data sources. These data sources include the CIA's World Factbook[76], the Best Buy catalog[77], the United States Geological Survey[78] for instance. The engine computes the results thanks to the Mathematica platform (handling algebra, statistics, visualization and more) which is another product from Wolfram research[79]. When a user enters a query Wolfram Alpha computes directly the answer and retrieves it through the most adapted data structure and visualizations.

Another major and recent initiative in the field of semantic search is the **Face-**

[70] http://www.crunchbase.com/company/powerset
[71] http://vimeo.com/994819
[72] http://www.apple.com/ios/siri/
[73] http://semanticweb.com/how-watson-works$_b$28437
[74] https://www.youtube.com/watch?v=rxU1Pg-80as
[75] https://www.wolframalpha.com/
[76] https://www.cia.gov/library/publications/the-world-factbook/
[77] http://www.bestbuy.com/
[78] http://www.usgs.gov/
[79] http://www.wolfram.com/mathematica/

book Graph Search[80], introduced by Facebook in March 2013 and currently available in English language. The Facebook social graph is a gigantic graph. More than a social graph it is fundamentally a graph of entities. It is highly multidimensional i.e. composed of a wide variety of objects and properties including people, places, companies and their relations. These entities are typical social objects to which the users subscribe and interact about. The Facebook Graph Search bar helps the user to build queries thanks to auto-completion. The queries are in the form of incomplete graph patterns e.g. *"peoples (the variable) who live in Paris and who like Claude Monet*, see figure 3.12. The user can then filters the results thanks to facets such as *gender*, *relationship*, *hometown*.

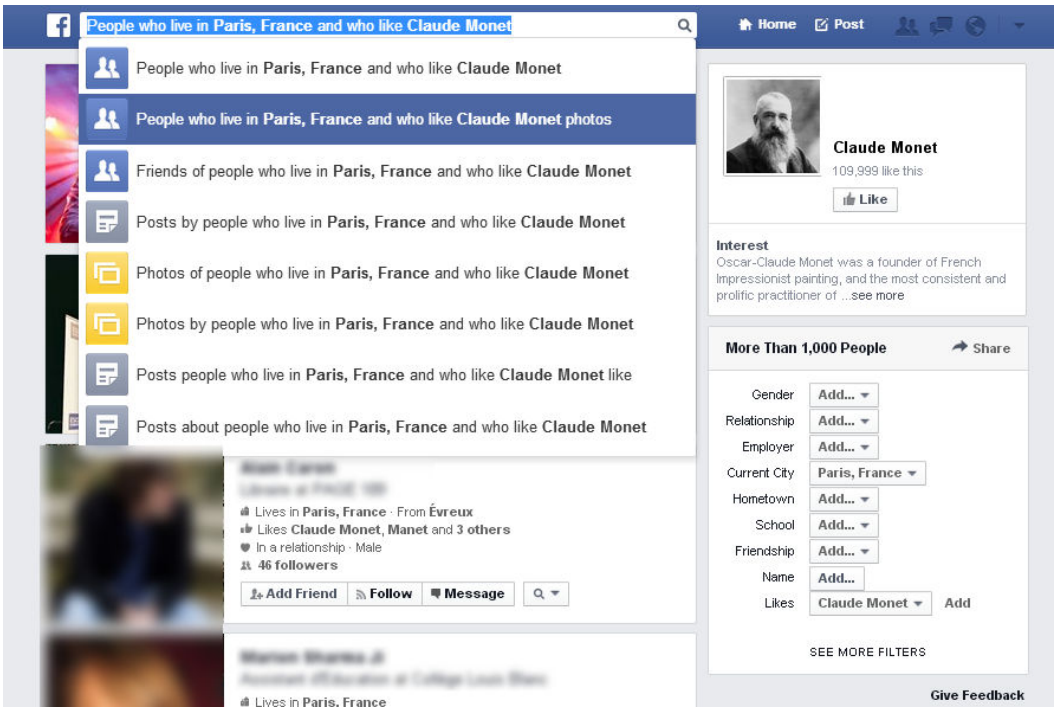

Figure 3.12: Example of a Graph Search query involving several types of objects and properties

An area of semantic search application is the e-commerce[81]. In this case the semantic search technologies are employed to rank the products and perform recommendations. The most visible and promising initiatives in the fields of semantic search comes from the traditional search engine and more particularly from Bing, Google and Yahoo. They launched functionalities based on semantic graphs later than many academic research prototypes because they need to reach a very high quality in terms of information retrieval and user experience. The quality of the results retrieved is a major economic stake for such players. Degrading the users'

[80] https://www.facebook.com/about/graphsearch

[81] http://techcrunch.com/2012/08/30/in-battle-with-amazon-walmart-unveils-polaris-a-semantic-search-engine-for-products/

perceived quality of the service through important technological shift is risky in the competitive market of search. Thus the fact they base more and more services on semantic technologies prove a relative maturity of the research in this field.

Signals of the major search companies interest for semantic search appeared since 2007. In 2007 Marissa Mayer, who was the Google's vice president of search products and user experience at the time, declared during an interview: *right now Google is really good with keywords, and that's a limitation we think the search engine should be able to overcome with time. People should be able to ask questions, and we should understand their meaning, or they should be able to talk about things at a conceptual level. We see a lot of concept-based questions not about what words will appear on the page but more like "what is this about ?" A lot of people will turn to things like the semantic Web as a possible answer to that"*[82]. In 2010 Google acquired Metaweb[83], a company founded in 2005 which is developing Freebase: a collaborative semantic database. At the time of writing Freebase contains more than 43 million topics and more than 2.4 billion facts. In 2012 Google was one of the initial funders of the Wikidata[84] project together with the Allen institute for artificial intelligence [85] and the Gordon and Betty Moore[86] foundation. They were later joined by the main Russian search engine Yandex[87]. Wikidata is a collaborative knowledge base built in the spirit of the Wikimedia projects i.e. open source, free license, community-driven, multilingual [187]. The objective is to identify, merge the structured data already present in Wikipedia as well as to clean and complete it through a community effort. Wikidata follows the linked data standard for data publication and is consequently part of the linked data cloud. It is a successful initiative that resulted in more than 30 million statements (triples) and more than 108 million edits as of February 2014.

In 2008 Microsoft acquired several data search technologies including natural language entity extraction when it bought the FAST Search and Transfer company[88]. Bing also contracted a partnership with the Britannica encyclopedia[89]. They used their data to provide rich snippets for the search results. It could also be a way to bootstrap the creation of their knowledge graph, like Google did with Freebase. As mentioned previously in this chapter Microsoft also acquired Powerset in 2008.

In 2009 researchers from Yahoo published a paper entitled "*a web of concepts*"[42]. In this paper the authors envisions search powered by semantic graphs *"for enabling many powerful applications, including novel search and information discovery paradigms"*. More particularly they detail their views about:

[82] http://www.infoworld.com/t/data-management/google-wants-your-phonemes-539?page=2,1
[83] http://www.freebase.com/
[84] http://www.wikidata.org/
[85] www.allenai.org
[86] www.moore.org
[87] http://www.yandex.ru/
[88] http://www.crunchbase.com/organization/fast-search-transfer
[89] www.britannica.com

- The usages: including optimization (understanding users' queries and content), augmenting web search or direct concept search, advertizing applications.
- The semantic graph-building techniques through web-scale extraction process, relational classification, aggregation mining.
- The challenges: including data noise, concepts matching, graph organization, maintenance, obsolescence, dynamicity.

In this paper the authors affirm: *"our goals are closely related to the semantic web, and we see the two approaches as synergistic. Our emphasis is on taking what exists on the web today and interpreting it and enabling richer applications (in particular, search), whereas the semantic web approach is to empower authors to publish content in a more interpretable form."*. Two years later Yahoo researchers gave a talk at the ISWC2011 conference industry track[90] entitled *Building a Web of Objects at Yahoo!*. It presented the Yahoo Knowledge Graph building progresses. Observing the slides[91] used during this presentation we notably learn that:

- The data is ingested from web extraction, feeds, editorial content.
- The data integration is done using Hadoop clusters (schema matching, object reconciliation, blending).
- The data are enriched thanks to social and behavior insights.
- Data quality assessment techniques are employed.
- The ontology has been developed over 1.5 years (at the time) by Yahoo's editorial team, it included 250 classes and 800 properties aligned with schema.org.

After this publication the company continued to sporadically communicate about this shift toward a *web of objects*[92].

The launch of the major search engines semantic search solutions happened during 2012 and 2013. In august 2012, Google publicly unveiled its Knowledge Graph[93]. It is a semantic graph that contained 500 millions objects and 3.5 billion facts (triples) when it was released. At the end of 2012 the Knowledge Graph was updated and available in Spanish, French, German, Portuguese, Japanese, Russian and Italian. The graph grew to 570 million entities and 18 billion facts in seven months. It is based, among others, on data sources like Wikipedia, Freebase and the CIA World Factbook. Shashi Thakur, the technical leader of the Google Knowledge Graph, affirmed[94] that the engineers' team didn't want to reuse the collaboratively edited schema of Freebase. Instead they analyzed the Google query stream

[90] http://iswc2011.semanticweb.org/program/industry-track/

[91] http://iswc2011.semanticweb.org/fileadmin/iswc/Papers/Industry/WOO_ISWC.pptx

[92] searchengineland.com/yahoo-were-moving-from-web-of-pages-to-web-of-objects-19524

[93] http://www.google.com/insidesearch/features/search/knowledge.html

[94] http://arstechnica.com/information-technology/2012/06/inside-the-architecture-of-googles-knowledge-graph-and-microsofts-satori/

in order to infer the main properties of interest associated to a resource e.g. casting for a film, height for a building.

The Google Knowledge Graph is a proprietary knowledge base and does not provide a public access. Consequently the list of objects that composes it is unknown. The first feature based on this technology is an augmentation of the search results by structured data. When the users enter a query an entity recognition is performed and structured information about the recognized entity are provided in a **knowledge panel**. For instance for the query "Claude Monet", the search engine displays in a structured format his birthday, birth place, parents names and more, see figure 3.13. The interface also shows his main artworks and a recommendation of others artists presented in the form of a carousel. With the knowledge panel the search engine performs now a combination of documents and data retrieval. A metric confirming the efficiency of the Knowledge panel is that since its introduction the traffic of Wikipedia decreased in all the languages covered by the Google Knowledge Graph[95]. This raises many questions as the Knowledge graph pulls a part of its knowledge from Wikipedia. The knowledge panel continues to evolve, it has been integrated to the mobile search interface and offers comparisons and filters functionalities[96]. Now it also includes statistics for some queries e.g.*"India population"*. It is also used to improve individual search results by providing information about the site the result comes from.

Among the major search engines Google was a pioneer in the deployment of search functionalities powered by such knowledge graph. It was also the first to largely communicate about it, presenting this technological asset as a major improvement and an important step toward the search of the future. The company insisted in the higher capacity of their search solution to *"understand"* the query and the results. The Google's main competitors reacted quickly.

Bing introduced its knowledge Graph, called Bing Satori (*understanding* in Japanese), and related functionalities in March 2013. They started to communicate about it in a blog-post entitled "*Understand your world with Bing*"[97]. The first functionality based on Bing Satori is called Snapshot and displays structured data about entities that are identified in the users' queries. It is close to the Google Knowledge Panel but integrates more social information (coming from LinkedIn, Twitter, Facebook). Another important difference with the Google approach is that Bing, often referred to Bing as a "*do engine*" by Microsoft. It put emphasis on the actions associated to entities. For instance the action *reserve*, *look at the menu*, *see the reviews* is often associated to restaurants. Bing processed the queries stream during several months for mining such associations, according to the director of Microsoft search Stefan Weitz[98]. Bing tries to propose the maximum of relevant

[95]http://stats.wikimedia.org/EN/ReportCardTopWikis.htm

[96]http://techcrunch.com/2013/09/26/google-improves-knowledge-graph-with-comparisons-and-filters-brings-cards-to-mobile-search-adds-cross-platform-notifications/

[97]http://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/

[98]http://arstechnica.com/information-technology/2012/06/inside-the-architecture-of-googles-knowledge-graph-and-microsofts-satori/

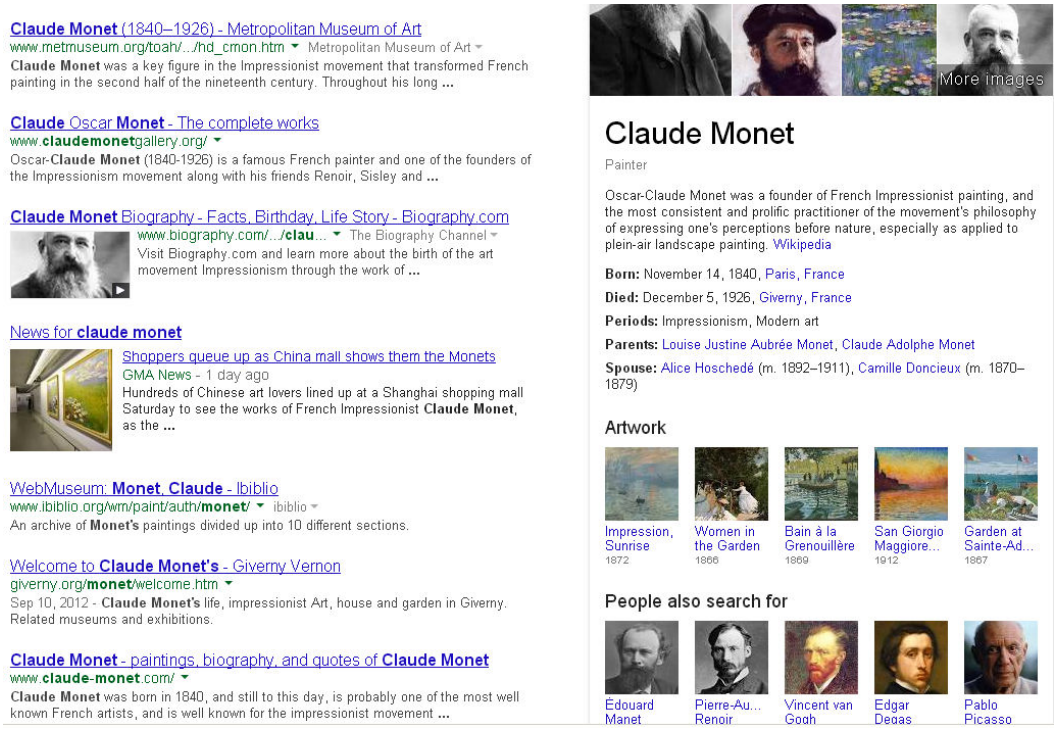

Figure 3.13: Google results for the query "*Claude Monet*", structured information are displayed on the left, in the Knowledge panel

actions associated to the search results. This approach is in line with the extension of schema.org to actions. The process of pairing actions to entity-bearing queries was described in [113]. Derrick Connell, the Microsoft's Bing experiences group leader, recently confirmed this *action-oriented* exploitation of Bing[99]. He affirmed that the integration with partners such as Yelp, OpenTable or TripAdvisor has to continue and that in the future the users might perform the actions (e.g. make a reservation in a restaurant) directly on the Bing results page. The company has also plans to partially open its graph to more third-party services, with the consequence of augmenting the number of actions available through Bing. The company has also developed natural language queries answering functionalities thanks to Bing Satori. The Satori knowledge graph is continuously updated e.g. doctors, lawyers, dentists and real estate properties[100], famous people[101] as well as food and drugs[102]. Contrary to the Google Knowledge Graph, Bing is only available in English at the time of writing.

After publishing papers about their vision in 2009 and giving information about the technical realization in 2011 Yahoo published a paper after the deploy-

[99] http://techcrunch.com/2014/03/30/microsoft-has-big-plans-for-bings-entity-engine/
[100] http://www.bing.com/blogs/site_blogs/b/search/archive/2014/03/31/150-million-more-reasons-to-love-bing.aspx
[101] http://searchengineland.com/bings-satori-adds-timeline-data-500k-famous-people-184969
[102] http://searchengineland.com/bing-expands-snapshot-new-food-drug-entities-191742

ment of their Knowledge Graph at the ISWC2013 in-use track[103]: "*Entity Recommendations in Web Search*" [20]. It is especially interesting as Yahoo, contrary to Bing and Google, is the only major search engine that has published such research papers.

This paper first offers an overview of the Yahoo Knowledge Graph and gives explanation on its elaboration. The core of the publication is focused on SPARK, a recommendation engine powering the Yahoo equivalent of the Google Knowledge Panel and Bing Snapshot. It also describes the evaluation performed to optimize and validate the new versions of this system. Some parts of the paper are voluntary vague and some values are undisclosed due to the confidential nature of the technology. The Yahoo Knowledge Graph is built from public data sources as well as data obtained from paid providers. More precisely the data sources include Freebase or Wikipedia (extracted using the DBpedia framework and proprietary tools), structured feeds and domain-specific sources. They are constantly monitored to update the graph in order to guaranty its freshness. The knowledge acquisition process is distributed and run on Hadoop. The paper confirms that the ontology is constituted of 250 classes and 800 properties and that was designed by the Yahoo editorial team. To improve its quality Yahoo use editorial curation as well as knowledge reconciliation techniques (*"record linkage, co-reference resolution, link discovery"*). Till today the Yahoo Knowledge Graph is voluntarily focused on the news, movies, television, music, sport and geography (points of interest) domains that are of interest regarding the search activity of Yahoo. The composition of the graph is shown on Figure 3.14. It is currently composed of approximately 3.5 million of entities and 1.4 billion of relations.

Domain	# of entities	# of relations
Movie	205,197	9,642,124
TV	88,004	17,126,890
Music	294,319	77,673,434
Notability	585,765	89,702
Sport	75,088	1,281,867,144
Geo	2,195,370	4,655,696
Total	3,443,743	1,391,054,990

Figure 3.14: Composition of the Yahoo Knowledge Graph

The search-engine knowledge graphs can be considered as private and closed linked data datasets. Unfortunately they do not provide any public access (e.g. APIs or dumps) to the public. They are not part of the linked data cloud or semantic web. One reason is that some of the data used to build these graphs are probably proprietary and can not be publicly unveiled due to commercial agreements. But the main reason is of course that these knowledge graphs are major competitive assets. An attempt to replicate the Knowledge Graph thanks to crowd-sourcing

[103]http://iswc2013.semanticweb.org/content/accepted-papers

techniques existed [175] but the company managed to stop it quickly[104]. At the beginning of 2013 Larry Page affirmed that Google was still at *"1% of where (they) want to be with the Knowledge Graph"*, confirming that is a key technology for the company. Moreover such semantic graph and related processing are critical for the search service but can also serve other services owned by these companies. Google uses Freebase to automatically annotates the videos by processing text metadata as well as video context (outside or inside the platform e.g. comments) analysis (title, description, tags), audio and video processing. In order to augment the quality of the annotations they use undisclosed techniques of weighting, co-occurence identification and human evaluations feedback[105]. Platforms like Youtube have critical content organization concerns that can be addressed by semantic technologies. In the same spirit Bing's shopping and travel sites make use of Satori's entities. The functioning of these recommenders is partially presented in chapter 4, to the extent of the available information.

3.6 Conclusion

In this chapter we showed that the publication of structured data on the web is an important and impactful trend. The major web players, the search engines and social networks in particular, integrate more and more structured data in their application. Their visibility within popular applications constitutes a motivation for web developers and a factor of traction. A lot of semantic data are embedded in web pages thanks to microformat, microdata and RDFa vocabulary. Meanwhile the linked open data cloud is growing and its central dataset, DBpedia, is already used in several end-users applications. The LOD cloud can be considered as the first deployment wave and the main achievement of the semantic web today. The incorporation of these structured data in search is called semantic search. The research on such systems exists since the 1980s; the semantic web vision catalyzed it. Today a lot of semantic search systems exists, they vary a lot in their objectives, their interfaces, the data they use and how they process it. The major web players start to deploy semantic search functionalities that have an important and visible impact on their applications. The users are increasingly familiar with the use of structured data in search, which is just at its beginning. The incorporation of structured data in search offers unprecedented possibilities to solve complex information needs. It is notably promising in the context of exploratory search where such semantics can improve both the information retrieval and the human computer interaction aspects of the systems. In the following chapter we propose a state-of-the-art review of the exploration and discovery approaches based on semantic data.

[104] http://openknowledgegraph.org/
[105] https://www.youtube.com/watch?v=wf_77z1H-vQ#t=52

CHAPTER 4

Linked data-based exploration and discovery

Contents

4.1 Introduction

The chapter 2 stressed the need to build and popularize efficient exploratory search systems. Alternatives are needed to complete the actual popular search solutions that are optimized for lookup tasks. The chapter 3 underlined the new possibilities brought by semantics incorporation in search systems. It notably allows to design novel search experiences in order to solve complex information needs. Supporting exploratory search tasks with the help of structured data is promising and

under research. The major search engines notably based their first (embryonic) exploratory search functionality on a knowledge graph. The semantics help to *go beyond the keywords* by interpreting the users' intent, assisting the exploration, explaining the results and more. Matching exploratory and semantic search is an inspiring idea but a number of issues have still to be addressed as both semantic search and exploratory search are relative immature research fields.

The linked open data cloud is the largest source of public and structured data today. Several appealing applications demonstrated that some LOD datasets are mature enough to serve as background knowledge for end-users purposes. An additional motivation for the research community is that linked data-powered applications lower the technical barriers to interact with the LOD and demonstrate its utility. The web of data has not been initially created to be navigated by humans and it needs to be exposed and interacted with in an intuitive manner. Thus, successful semantic search functionalities and applications constitute a factor of traction for both the data publication and the applications' development. They are a necessary interface between the casual users and the linked data cloud. However, even the experts need tools to obtain good mental representations of increasingly large and heterogeneous linked datasets today.

Supporting exploration and discovery thanks to linked data is both inspiring and challenging. Both the information retrieval and human computer interaction aspects have to be researched. Nowadays the relative immaturity of the field leads to a profusion of heterogeneous approaches, systems and evaluations. All these contributions address the challenge of hiding the semantic data complexity to the users. Presenting the data in a meaningful and appealing way is crucial. It requires a difficult selection and prioritization process in the context of highly connected and heterogeneous linked data graphs. A central question when displaying the web of data is to determine what has to be shown to the users. In other words the system developers have to determine what combination(s) of triples constitute(s) the result unit satisfying the users' information need. Allowing complex interactions with the data without being aware of the underlying query mechanism, data model and structure is another major concern. Generally the tension between the data display, the interactions expressiveness and the interface intuitiveness is high and drives the design choices. Such choices result in systems that implicitly target different types of users, having different levels of expertise.

In this chapter we will review the linked data based exploration and discovery approaches, within broad areas of classification. More particularly we review (4a) the linked data browsers and their variants, (4b) the semantic-similarity and relatedness based approaches including in particular the recommenders, (4c) the linked data based exploratory search systems. The systems are ordered chronologically in each subsection. A short version of the state-of-the-art review and the corresponding analysis were published in [125].

4.2 Linked data browsers

One of the first generation of tools designed to explore linked data was semantic browsers. Before their existence it was necessary to read the serialized RDF files to discover and understand the data. The first semantic browsers were strongly inspired by the web pages browsing and allowed the users to navigate into the linked data space in a *one-resource-at-a-time* mode, often by following the currently displayed resource outgoing property. Numerous systems were conceived and employed diverse approaches for lowering the visualization and interactions complexity. The display of the graph as it is has only a minor interest for end-user applications and further semantic-based processing is needed to obtain a comprehensible and appealing navigation [89]. In this state-of-the-art survey we review the semantic web browsers according to a broad classification. Our first category is the text-based browsers. Our second category is the visualization-based browsers that use and potentially combine visual presentation(s) such as graphs, images, maps and timelines. Our third category is composed of faceted browsers. Faceted browsing is a successful interaction model that is particularly efficient for semantic data exploration. This interaction mode enables sorting and filtering the results thanks to their semantics. The fourth category named *other browsing paradigms* reviews innovative and singular browsing approaches enabled by linked data.

4.2.1 Text-based browsers

Text-based browsers use textual structures such as tables and lists to present the data. In such systems when a resource is browsed the system often displays its outgoing properties and associated objects/values. They are often the simplest and the earliest systems and do not provide major semantic-based support for easing the data understanding and interactions

Noadster[1] [162] is an early (2005) and generic RDF browser that performs a property-based clustering on the data. The most prevalent properties are declared as *"stop properties"* and are not considered during the clustering process. The property-based clusters are used to structure the results: frequent properties appear higher in the results-tree.

Disco[2] (2007) is a simple server-side browser that renders the RDF data in columns of property-value pairs associated to the currently browsed resource (object) in a table. It lists the data provenance at the end of the page and allows the navigation from one URI to another on distributed data sources by dereferencing.

[1] http://homepages.cwi.nl/ media/demo/noadster/
[2] http://wifo5-03.informatik.uni-mannheim.de/bizer/ng4j/disco/

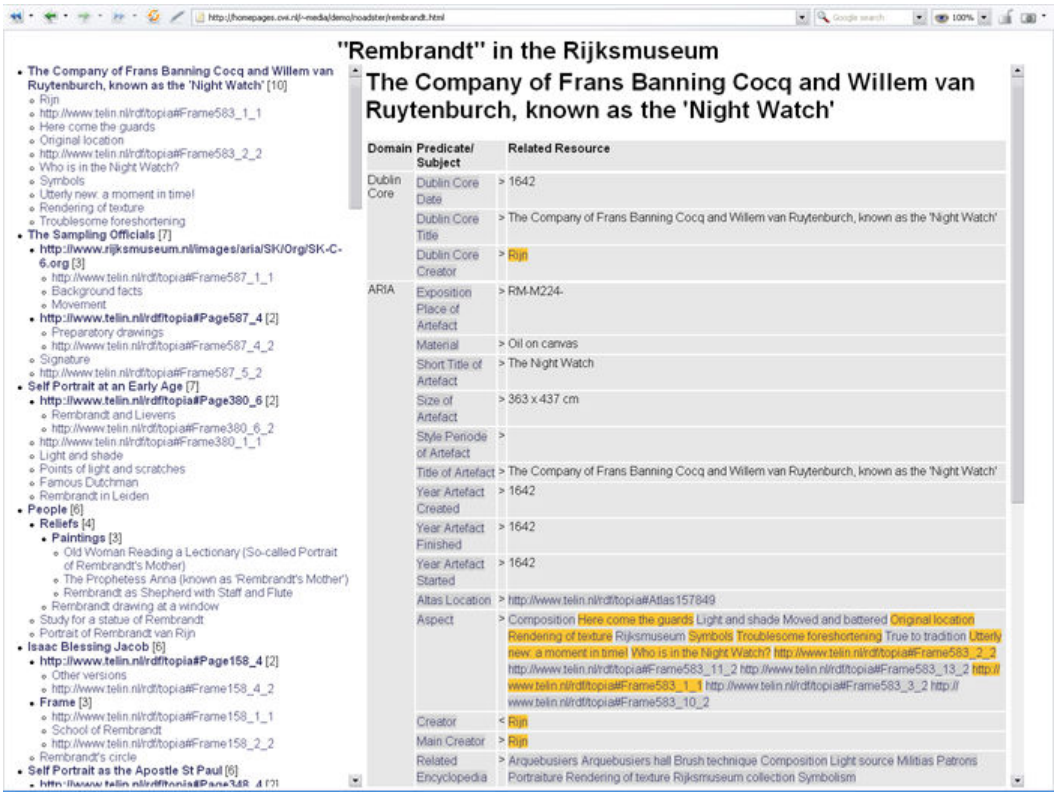

Figure 4.1: Noadster

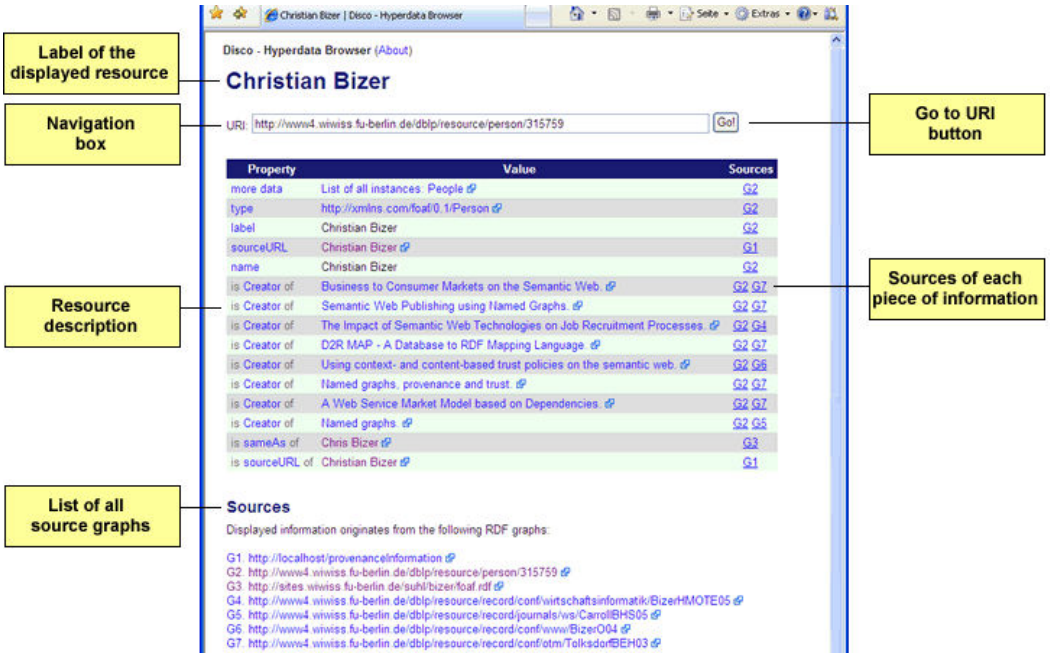

Figure 4.2: DISCO

Marbles[3] (2007) is a server-side browser, also available in the form of a local application, that formats the RDF triples using Fresnel (*a simple, browser-independent vocabulary for specifying how RDF graphs are presented*[4]). Marbles takes advantage of the distributed aspect of the LOD cloud. It retrieves additional data about the browsed resource from Sindice[5] (semantic index), Falcons[6] (semantic search engine) and Revyu[7] (semantic reviews site). It also dereferences the *owl:sameAs* and *rdfs:seeAlso* properties to retrieve extra knowledge. Colored bubbles (the *"marbles"*) help the users identify the sources of the data retrieved. Marbles also provides a SPARQL endpoint for querying.

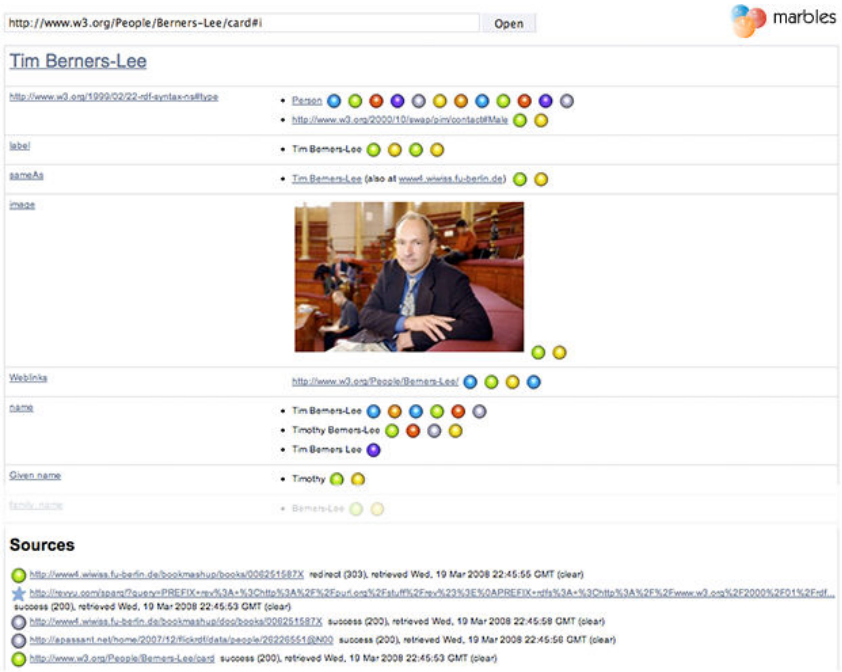

Figure 4.3: Marbles

URIburner[8] (2008) is a semantic browser that renders the data in property-value pairs. It supports dereferenciation and embeds images and web pages in its presentation. URIburner allows to export the data in several format including CSV, the RDF syntaxes, JSON and microdata. Additionally it offers alternative views on linked data like graph ones as well as manual query and query-by-example mechanisms. URIburner provides a keyword-search functionality, within a results set or as a starting point for retrieving and browsing linked data.

[3]http://wiki.dbpedia.org/Marbles
[4]http://www.w3.org/2005/04/fresnel-info/
[5]http://sindice.com/
[6]http://ws.nju.edu.cn/falcons/submituri/index.jsp
[7]http://revyu.com/
[8]http://linkeddata.uriburner.com/

Sigma[9] (2010) standing for *Semantic Information MAshup* [181] retrieves triples from a large number of distributed data sources starting from the string or URI entered by the users. Each Sigma results page is a consolidated entity description obtained thanks to large-scale indexation, advanced reasoning and aggregation strategies at data and ontology levels. The users can interact with the results: confirm or dis-confirm their relevance, filter and reorder them. Sigma proposes the results exportation in RDF, JSON and Really Simple Syndication (RSS)[10] formats.

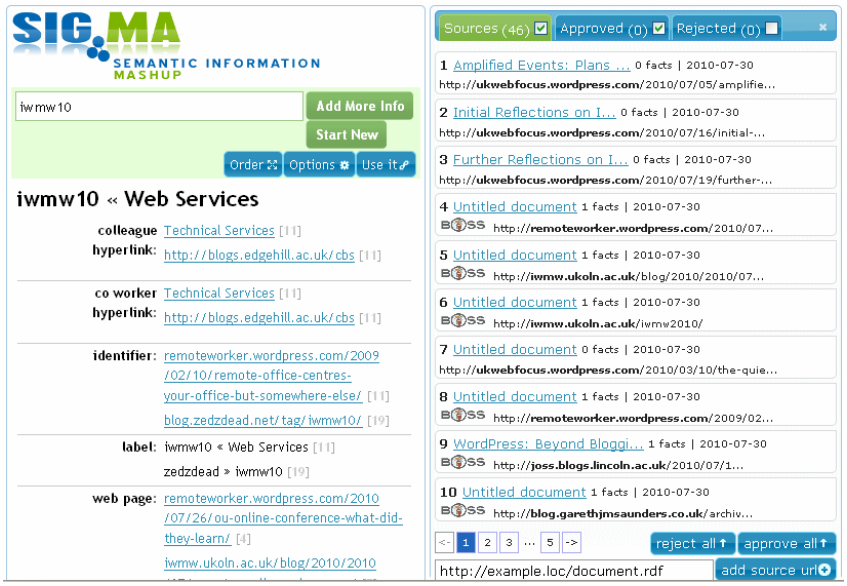

Figure 4.4: Sigma

4.2.2 Visualization based browsers

When browsing linked data, visualization features can significantly lower the users' cognitive load. They help them overcoming the data complexity by relying on their natural perceptual ability. Recently several personalized visualization approaches based on templates appeared. Such templates aim to let the users design the visualizations that are the most adapted to their information needs.

IsaViz[11] (2001) is *"a visual environment for browsing and authoring RDF models represented as graphs"*. It is an early initiative of the W3C to ease and promote the use of RDF. It proposes a 2.5D interface and helps the users in the creation and the edition of RDF graphs. IsaViz also offers import and export functionalities in common RDF syntaxes. Zooming, search and filters are available and ease the visualization and interaction with data.

RDF Gravity[12] (2004) is an RDF and OWL visualization tool [60]. It proposes

[9] http://sig.ma
[10] http://www.rssboard.org/rss-specification
[11] http://www.w3.org/2001/11/IsaViz/
[12] http://semweb.salzburgresearch.at/apps/rdf-gravity

filtering, search and allows the users to reposition the nodes. Zoom and panorama functionalities are also available. The resources are displayed in different colors according to their types for a more comprehensive display. RDF Gravity also supports two structured query languages: SPARQL and the non-standard RDQL[13].

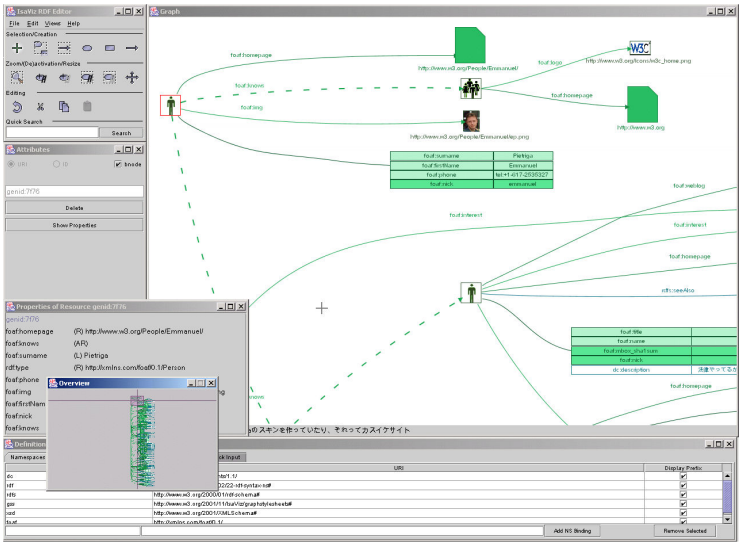

Figure 4.5: Isaviz

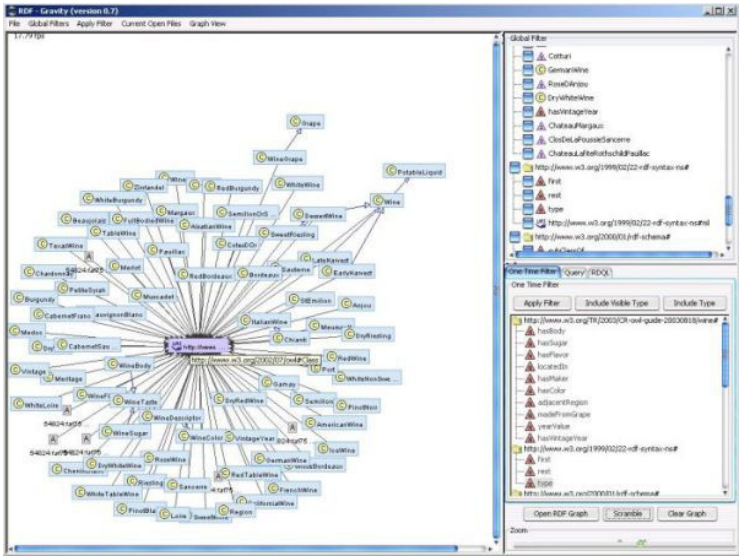

Figure 4.6: RDF gravity

Tabulator[14] (2005) is an RDF browser that displays the data hierarchically through a tree-view [14]. By clicking the hierarchical trees the users browse

[13]http://www.w3.org/Submission/RDQL
[14]http://dig.csail.mit.edu/2005/ajar/ajaw/tab

through increasing levels of details. The starting point is an entered URI or a SPARQL query. The SPARQL queries are stored in order be re-executed when needed. In the last version the users can edit the data through the interface. Geo-tagged resources can be shown on a map and time-stamped resources can be displayed on a calendar or a timeline. One particularity of Tabulator is that the RDF data are stored in a client-side triple store. This approach led to important scability problems. The solution of query-able server-side triple stores is now the most common.

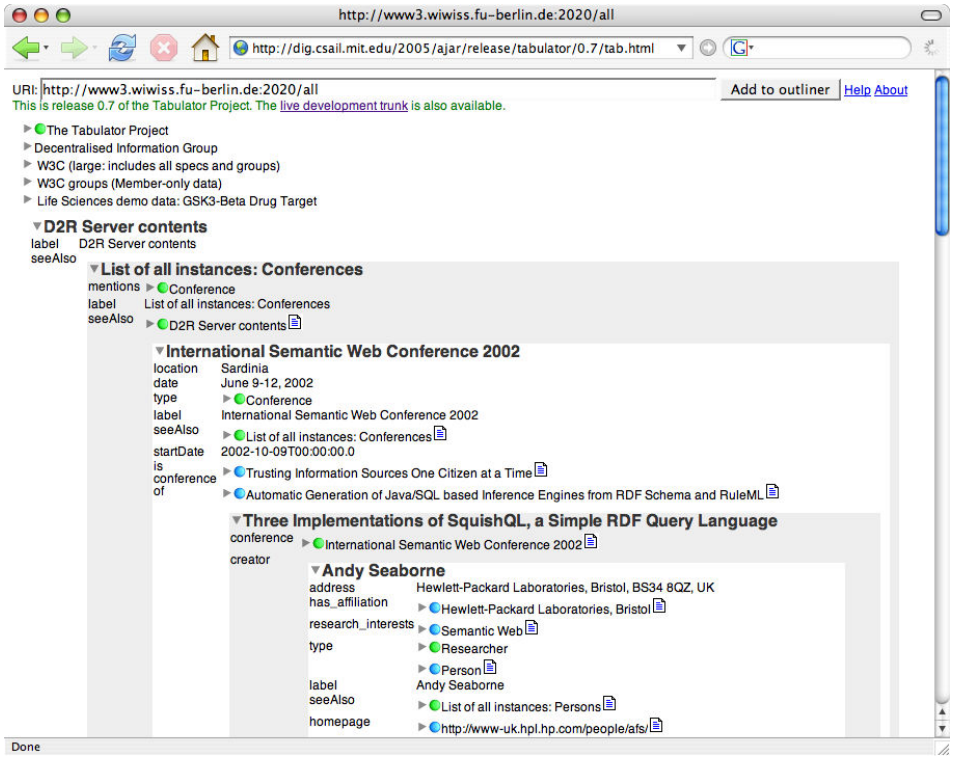

Figure 4.7: Tabulator

DBpedia Mobile[15] (2007) is a location-aware mobile web application [13]. It displays the geo-tagged DBpedia resources surrounding the user (or elsewhere) on a map, namely Open Street Map[16]. The application uses the GPS phones capacities and processes the DBpedia *geo:lat* and *geo:long* properties of geo-tagged resources. When the users click on them they are displayed in a mobile version of the Marbles browser. Filters and context-aware strategies are used to minimize the amount of displayed triples.

[15] http://beckr.org/DBpediaMobile

[16] Open Street Map is an openly licensed map of the world being created by volunteers using local knowledge, GPS tracks and donated sources: http://www.openstreetmap.org/

Figure 4.8: DBpedia Mobile

Fenfire[17][71] (2008) is a linked data browser application that displays the neighborhood of the browsed resource in a graph view. It supports dereferenciation. Fenfire processes the *rdfs:seeAlso* properties to retrieve additional knowledge and the *rdfs:label* to display the resources' names. The users can switch to a list view if needed. They also have the possibility to edit the RDF graphs and save them locally.

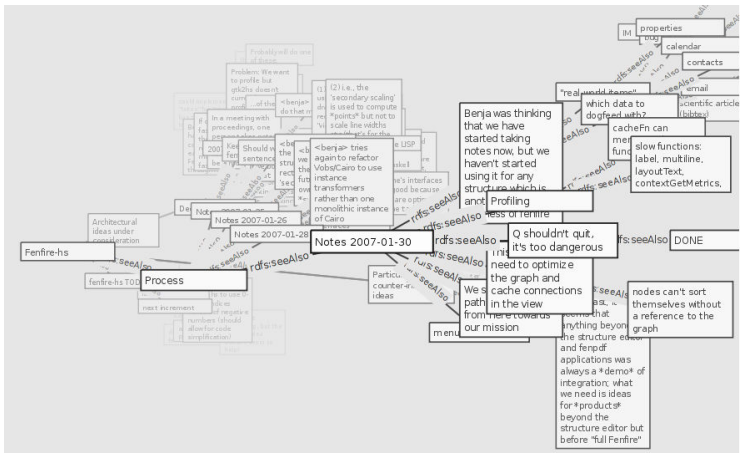

Figure 4.9: Fenfire

The OpenLink Data Explorer[18] (2008) is a browser extension offering a set of textual and visual views on the data. The visual views include:

- the *what* view showing a list of data provenance sources.
- the *where* view where geo-located resources are shown on a map.
- the *when* view displaying resources having temporal attributes on a timeline.

[17]http://fenfire.org/
[18]http://ode.openlinksw.com/

- the *who* view displaying only the resources representing persons using the FOAF vocabulary.
- the *tag* view proposing links to the del.icio.us[19] tags related to the currently browsed data.
- the *SVG graph* view displaying the data in the form of a graph.
- the *navigator* view grouping the resources by their types.
- the *custom* view allowing the users to select Fresnel-based visualization template.

LENA[20] (2008) offers a mechanism to build Fresnel-based visualization templates using SPARQL queries [97]. The lenses-template authoring is expressive and supports complex criteria. The objective is to propose different views corresponding to different users' interests and expertise.

LESS (2010) proposes a proprietary template language and editor [7]. It allows the users to create views on linked data thanks to the LESS Template Language (LeTL). The input data rendered by the template can be specified by entering URI(s) and/or executing SPARQL query(ies). The result can be output in HTML, RDF or Javascript and JSON. Such export can then be imported in another system (blog, wiki, application, etc.). The templates are shared in a collaborative repository to favor their reuses among the users' community.

In **Dadzie and al.** ([41], 2011) the authors stress that a single linked data browsing/display solution will never meet the requirements of all the users. Indeed, linked data are heterogeneous and used in a wide range of domains. To solve this problem the authors propose a template-based approach to display the data. The templates are associated to RDF classes (e.g. *foaf:Person*). Overview as well as a detailed view are proposed and the users have the possibility to interact with the visualizations.

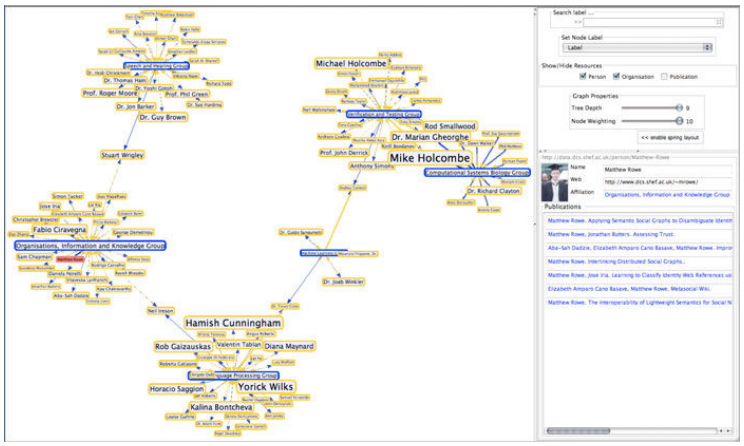

Figure 4.10: Template-based visualization approach presented in [41]

[19] https://delicious.com/
[20] http://isweb.uni-koblenz.de/Research/lena

4.2.3 Faceted browsers

Facets were defined as "*the categories, properties, attributes, characteristics, relations, functions or concepts that are central to the set of documents or entities being organized and which are of particular interest to the user group*" in [107]. The faceted classification was originally invented by the librarian Shiyali Ramamrita Ranganathan in 1959 [158]. He proposed to describe the books thanks to 5 facets to physically order them in the libraries: personality, matter, energy, space, time. Later faceted browsing, filtering and search became powerful interaction paradigms in computer science. A faceted search system *"presents users with key-value metadata that is used for query refinement"* [98]. When a facet value is selected the system constrains the results to the items that are compliant to this value. Each time a constraint is applied the available facets and facet values are updated, see Figure 4.11. Multiple constraints can be applied at the same time.

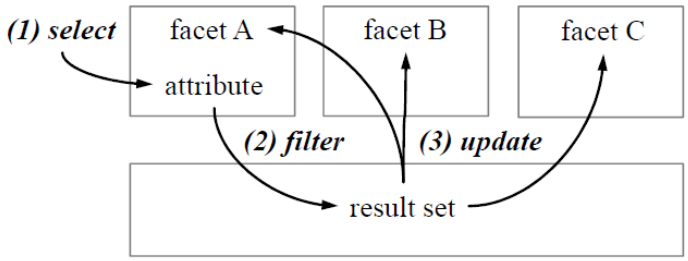

Figure 4.11: Faceted search functioning model, taken from [73]

Facets ease the result browsing by iterative drill-down (refinement) or roll-up (generalization) operations. It is particularly powerful during exploratory search as it gives *"the users the ability to find items based on more than one dimension, to see breakdowns and projections of the items along different axis, which helps users gather insights about the data they are exploring"* [2]. Generally it avoids *dead-end alley* by proposing only facets combinations that produce at least one result. Faceted browsing is also powerful to suggest unexpected and unknown browsing perspectives. At the same time a drawback is that it imposes an information structure that may limit free forms of exploration. [106] and [170] showed that relevant and meaningful information categorization has a positive effect on reflection, discovery, information finding as well as learning. In [102] the authors measured that, when available, the observation and manipulation of facets have a major role in the browsing process, accounting for approximately half of the time.

Facets help to overcome the users common *trial-and-error* querying tactic by introducing progressive refinement mechanisms. They have an active role by refining the queries and easing browsing. They also have a passive role by helping the users without a-priori knowledge to understand the results space through the exposition of important results characteristics. Facets are appreciated when the ranking fails to immediately retrieve interesting results. This problem may occur in an exploratory search context where some queries are *vague, broad and general*

[84]. They help the users find deeper (less well ranked) results [84] [104]. Facets can also be a major source of inspiration for users that explore an unknown domain. According to [129] they introduce several functions in an interface that significantly help during an exploratory search task:

- *"Vocabulary control*
- *Site navigation and support*
- *Overview provision and expectation setting*
- *Browsing support*
- *Searching support*
- *Disambiguation support"*

To go further previous research works showed that facets are efficient for supporting exploratory search. [153] compared the results of users executing the same search tasks with 3 systems: one with the results being ranked, another with the results being clustered and the last one with a faceted interface. The users found more results using the faceted system for exploratory search tasks (respectively an average of 5.60 vs. 4.53. vs. 7.80). There was no significant difference for a structured search task. In [204], the authors compared a baseline interface with a faceted one for images search. They noticed that the users faced empty results three times more with the baseline interface. The users evaluated the faceted interface *"easier to use, more flexible, interesting, enjoyable, simple, and easy to browse"* but *"a bit more confusing"*. The quasi-totality of the participants judged the interface more efficient than the baseline for the leaning tasks. In [184] the authors compared the feedback from 19 users about 2 interfaces (a baseline versus a faceted one) after the execution of 1 lookup and 2 exploratory search tasks. According to the users, the understanding and the perceived relevance of the results were more satisfying with the help of the facets. The most appreciated functionalities were the capability to switch from one facet to another, the previews of the results, the possibility of activating several facets at a time and the breadcrumb function. [206] compared three search tasks (one look-up and two exploratory ones) on two interfaces including a faceted one. The users were more satisfied with the faceted interface when executing the exploratory search tasks even if it was a bit less intuitive. A minority of participants also affirmed to be confused by the constant changing of facets along the navigation. [50] summarized the benefits of faceted interfaces found in empirical studies. Some of them, listed below, confirm their usefulness for exploratory search:

- Facets are useful for creating navigation structures.
- Success in finding relevant results is higher with a faceted system.
- Users find more results with a faceted system.
- Users prefer search results organized into predictable, multidimensional hierarchies.

- Participants' satisfaction is higher with a faceted system.
- Users are more confident with a faceted system.

It is difficult to compute and structure the facets from raw, unclassified and not curated information collections such as heterogeneous web pages [176]. On the contrary, structured documents and entities collections are privileged contexts to implement faceted search. The use of facets is especially popular on e-commerce web applications such as Amazon[21], see figure 4.12. These companies have a strong interest in guiding the users as quickly as possible to the products of interest in order to increase the sales. It is particularly adapted for such companies as they maintain a structured products database having well-defined characteristics. As these websites are very popular a large amount of web users are already familiar with faceted browsing to a certain extent. It constitutes a powerful alternative to visualization-based approaches that can be difficult to apprehend for many users.

The semantic web data model where resources are related to others resources through well-defined properties is interesting to set up a faceted interface. Roughly the results correspond to the triples' subjects, the facets to the properties and the facets values to the objects and literals. Setting faceted search and browsing systems on top of linked data has been an active topic of research. It can considerably ease the interactions with the high-dimensional graphs like the semantic web ones. By activating several facets at a time the users can implicitly build complex SPARQL queries through an intuitive interface in few iterations. The ontological structure inherent to semantic web data notably helps building facets hierarchies. Nevertheless the linked data complexity also brings several specific challenges regarding faceted search including the relevant facet identification, their ranking, their interdependence management, etc. Faceted search systems built on top of semantic data sources are presented hereafter:

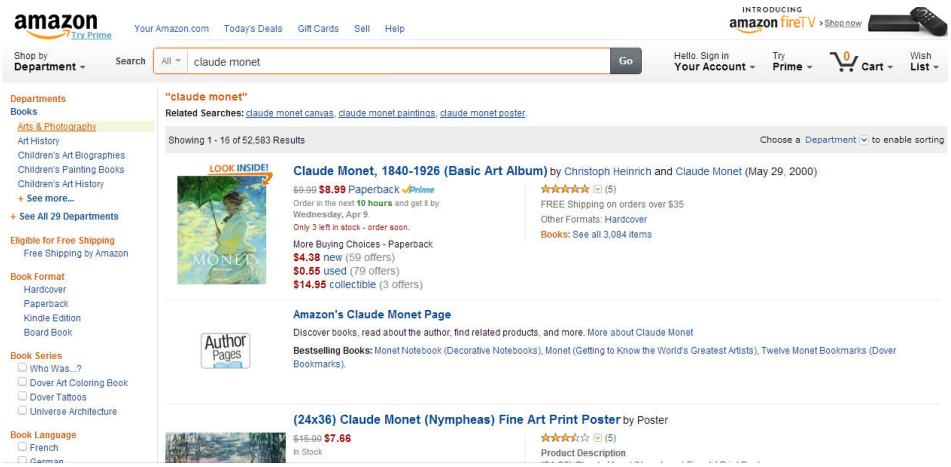

Figure 4.12: Facets proposed by Amazon for the query *"Claude Monet"*

[21] http://www.amazon.com

Piggybank[22] (2005) is a browser extension that aims to both publish and explore RDF data [80]. It was released at a time when semantic data were rare and sparse. The PiggyBank users can write screens-crapers template that convert the content of browsed web pages into semantic data. They also have the possibility to tag these generated data and to upload them in a shared triple store. Piggybank offers browsing capabilities and notably proposes several views on the data (list, calendar, graph, map, timeline) as well as facets filtering.

Longwell[23] (2005) is a web application for browsing and searching large RDF datasets that offers a faceted filtering mechanism. The facets and their values are heuristically derived from the dataset. These heuristics are pre-configured by the user using the *"facet configuration vocabulary"*. Thanks to the configuration files the users are able to specify which facets are available for a dataset and in which order. Longwell also proposes a free-text filtering functionality that constrains the results to the ones containing the searched string in their properties' values.

MuseumFinland[24] (2005) is a faceted semantic browser that was implemented on the top of a Finnish art-collections RDF knowledge base [83]. The latter was obtained by converting, aggregating and aligning 3 museums relational databases and using 7 ontologies e.g. artifacts, collections, materials ones. The knowledge base creation was semi-automatic as some curators added, edited and removed data. MuseumFinland allows the public and the art professional to browse and discover links in a unified way. The users rely on the hierarchically organized facets to filter the results. Preview counts help the users in the manipulation of the facets. The application also offers keyword search. The facets of the items are also shown on the individual result page, offering a traversal navigation capability.

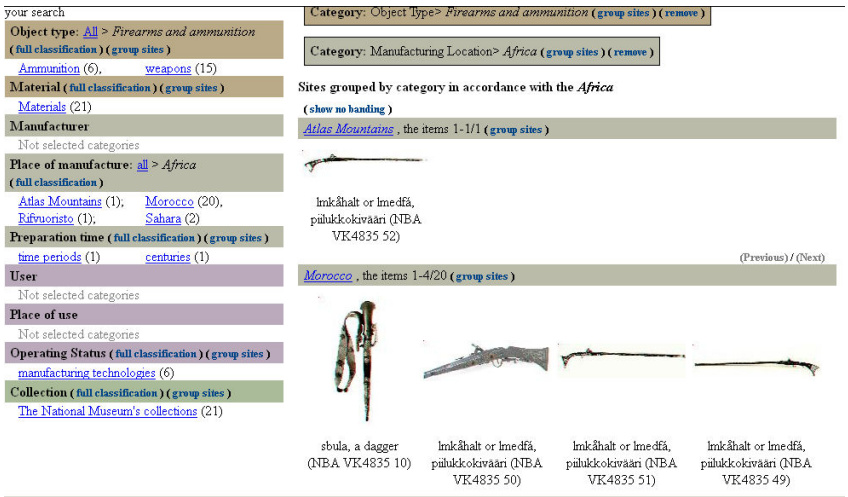

Figure 4.13: Museumfinland

[22] http://simile.mit.edu/wiki/Piggy_Bank
[23] http://simile.mit.edu/wiki/Longwell_User_Guide
[24] http://www.museosuomi.fi/

mSpace [196] (2005) is based on an interaction model that uses the facets dependencies as a support for browsing. mSpace leverages such dependencies to propose a variety of exploration mechanisms. The facets are organized horizontally (e.g. era, composer, place in the domain of classical music). The order is not neutral and constitutes a hierarchy, where the left-most column is the top level. The value(s) applied in each facet constrain(s) the displayed values of the facets on *the right*. The users can rearrange the information space by changing the orders of the facets. They can consequently place the facet they are the most comfortable with in the top position. The users have the possibility to swap, delete and add new facets. mSpace proposes multimedia preview cues (including sounds, videos) that help to understand the content of the facets and their values. Details about the currently selected results are displayed in a specific panel at the bottom. It is possible to save the discovered results as well as the current information space configuration for sharing and further reuse. The export of all the meta-data is possible in CSV, XSLT or plain text.

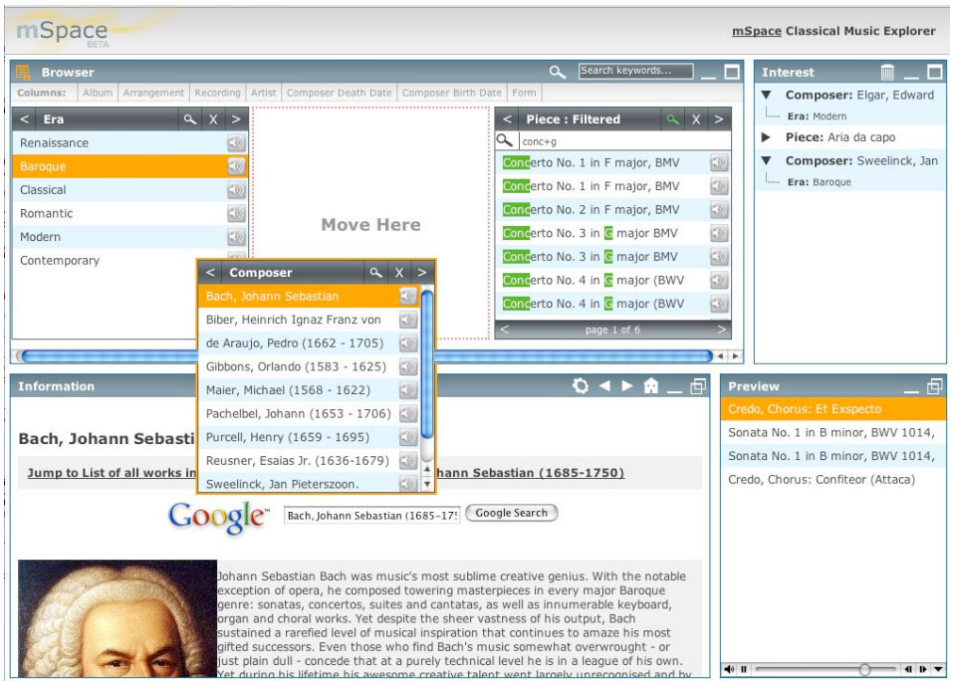

Figure 4.14: Mspace

/facet (2006) is a faceted browser over RDF data [78]. Its specificity is to automatically enable faceted browsing over large and heterogeneous linked datasets where manual configurations are too fastidious or unfeasible. The authors implemented /facet over a dataset aggregated from several museums databases with only minor alignments. The facets and their dependencies are generated with the help of ontological knowledge, especially the RDFS classes and their associated properties. Through the interface the users start to select the resource type (e.g.

Work) and filter the list with other facets corresponding to its related properties (e.g. authors, place of creation, artistic movement). The facets are presented in a hierarchy formed by the subsumption links. A full-text search is proposed for each facet in order to find quickly the desirable facet values. Specific views are available for some facets e.g. time-line. This automated process to set up faceted browsing has the advantage to be applicable on every linked database and easily takes account of knowledge base updates. Nevertheless relying extensively on the data model sometimes lacks of filtering and generates uninformative or hardly understandable facets for the users. Data inconsistencies, often present in composite datasets, are also exposed.

Figure 4.15: facet

BrowserRDF (2006) is a faceted browser prototype implementing several facets operators [142]. Some of them are complex and unusual. All the operators are formally described. The authors also propose an automated facet ranking algorithm. The authors have the objective to overtake two common limits of the faceted interfaces: first the lack of expressiveness in the facets operators, often limited to simple selection and their combinations, second the need of a manual intervention for the facets identification and ranking. In addition to the classic facets selection operator the authors formally define an *"existential"* selection (filtering the results according to the existence of related property/ies) and a *"join"* selection (two or more selections evaluated in conjunction). The inverse and intersection of these operators are also formally defined. The automated facets ranking technique has been developed to avoid manual intervention and is especially useful for large and heterogeneous datasets. The technique is based on three metrics that are agnostic

to the data:

- Predicate balance: *"Faceted browsing can be considered as simultaneously constructing and traversing a decision tree whose branches represent predicates and whose nodes represent restriction values. Tree navigation is most efficient when the tree is well balanced because each branching decision optimizes the decision power. We therefore use the balance of a predicate to indicate its navigation efficiency"* [142].
- Predicate frequency: *"A suitable predicate occurs frequently inside the collection: the more distinct resources covered by the predicate, the more useful it is in dividing the information space"* [142].
- Object cardinality:*A suitable predicate has a limited (but higher than one) amount of object values to choose from* [142].

The prototype was experimented and received good feedback. It did not implement the inverse facets operators. However the authors affirm that their facets ranking approach did not corresponded to users' expectancies and requires further research.

Exhibit (2007) is a lightweight framework for publishing structured data and setting an interactive interface without technical skills [82]. It offers an interface framework that provides views and faceted browsing on the published data. The views notably include thumbnails, maps and time-lines. They are configured by the data publisher e.g. display birth places or death places on a map. The facets allow the users to filter the data displayed in the views. Preview counts provide an extra-help during the use of facets. The Exhibit configuration files have a proprietary format and are written in JSON. The authors offer a web service to convert RDF data (and other formats) in the required format. The facets are manually declared by the data publisher.

Humboldt (2008) is a faceted browser that was designed to support exploratory search tasks over a local RDF graph [96]. Contrary to the majority of semantic web faceted browsers Humboldt shows facets that correspond to classes instead of property-based ones. For example in the cinema domain the facet *Person* appears instead of *directed, stars in, has produced, has written, has edited, and was awarded for*. Proposing class-based facets simplifies the interface by decreasing their amount. The users can therefore ask for more precision if needed. Humboldt also supports browsing through pivoting operations e.g. switching the focus from the *Film* facet to the *Person* facet. The results list will change to persons and the available facets will be recomputed. Thus the users can explore the whole dataset through pivoting operations. Humboldt supports implicit query-building as the selected facets remains activated along the successive pivotings. A history displays the interactions sequence and allows to modify the previous steps. The authors performed an evaluation with a cinema-related DBpedia subset. During a qualitative evaluation the participants expressed globally positive feedback about the prototype.

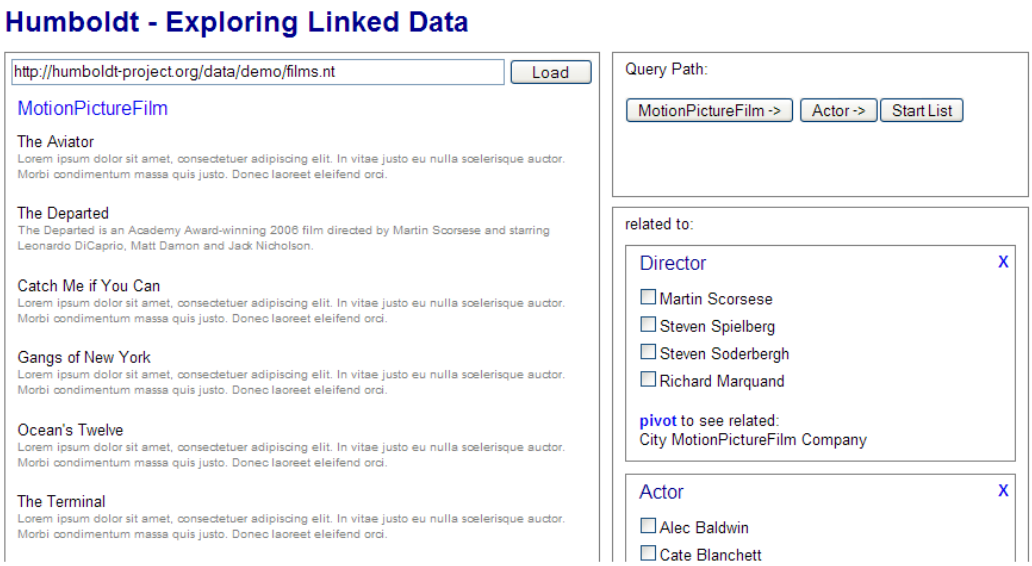

Figure 4.16: Humboldt

Visinav[25] (2008) is a semantic browser that proposes four atomic interactions [70]. By combining these four operations the authors are able to express complex queries without any technical knowledge:

- Keyword search: all the search sessions starts with a classic keyword search that identifies the set of objects that will be explored and refined with the others operations. The search is performed by matching the users query with the RDF literals.
- Object-focus: when the users click on a result it displays a detailed view showing its associated properties-values pairs.
- Path traversal: the users can express joint queries thanks to successive drags and drops e.g. objects created by people who know Jim Hendler.
- Facet specification: a classic faceted filtering mechanism is available.

All these interactions are formally described and are agnostic to the data. According to the authors faceted systems are often set on the top of manually crafted and limited datasets having a defined and controlled ontological schema. Thus the majority of proposed approaches do not scale in the semantic web context where there is a high variety of data and schemas that makes manual operations unfeasible. To demonstrate their approach the authors developed the Visinav prototype which operates on top of 18.5 million RDF triples aggregated from 70.000 sources. Various views (list, table, timelines and maps) and export functionalities are available.

[25] http://sw.deri.org/2009/01/visinav/

Faceted Wikipedia Search[26] (2010) is a simple faceted browser built on top of DBpedia [68]. The facets and their values are ranked according to their prevalence in DBpedia. The interface allows the users to explore and find answers to structured information needs that can be time-consuming using the traditional Wikipedia interface.

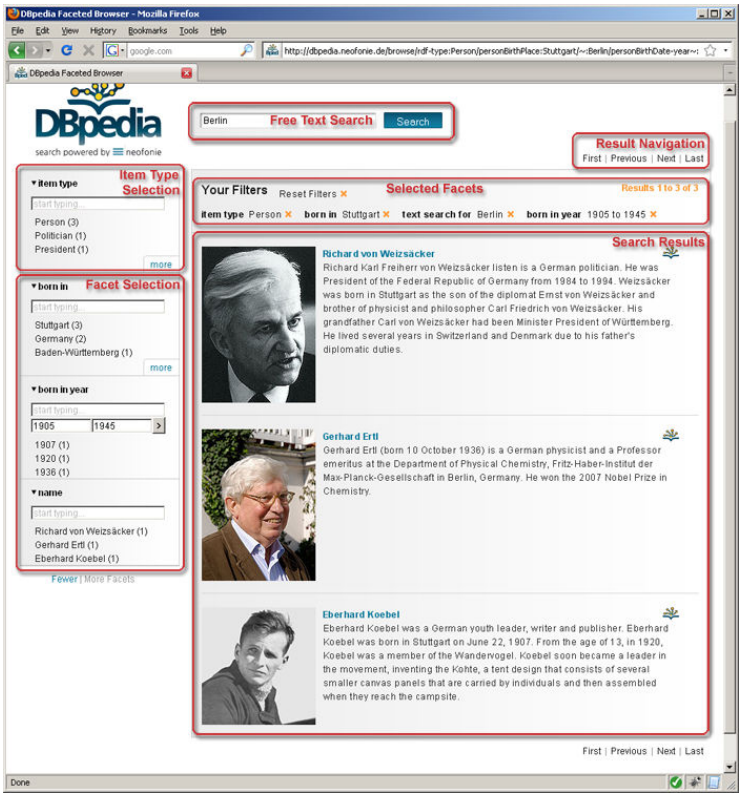

Figure 4.17: Faceted Wikipedia Search

4.2.4 Other browsing paradigms

The browsers presented in this section propose innovative and singular interaction paradigms implemented on top of linked data sources.

Parallax (2009) aims to overtake the classic *one-resource-at-a-time* interaction paradigm [81]. Indeed, the majority of web browsers allow the users to navigate one page at a time through the hyperlinks: the web page constitute the information unit with which the users interact. Several previously described text-based semantic browsers offers a similar interaction mechanism by presenting only the properties/values pairs associated to the currently browsed resource. The authors of Parallax introduce the *"set-based browsing"* paradigm that allows the users to browse several links at a same time i.e. from a set of entities to another set of entities. This concept is demonstrated thanks to a web application built on top of

[26] http://dbpedia.org/FacetedSearch

the Freebase knowledge base[27]. For example the users have the possibility to navigate from the set of American presidents to the set of all their children, or spouse, or political parties thanks to the corresponding semantic relations using a *"browse all"* functionality. Thus, the navigation does not occur through a single link but though a set of semantically similar links. Along the exploration the users can apply various constraints on the currently displayed data thanks to a faceted filtering mechanism. Visual views are also offered (e.g. maps, timelines and plots) to display and analyze the sets of results obtained from the successive interactions. A browsing history helps the users edit each interaction step e.g. modifying the previously activated facets. As the successive sets are dependent, a modification at a step leads to a re-computation of the subsequent ones.

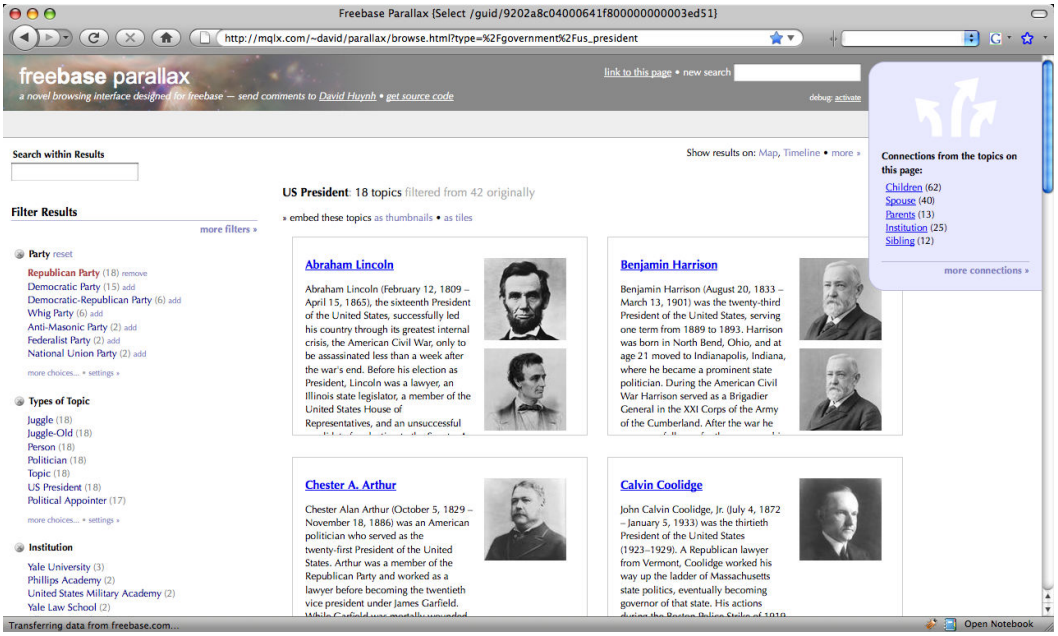

Figure 4.18: *parallax*

The set-based browsing paradigm was also implemented in a browser extension called **Companion**. This extension augments the classic web-pages browsing experience. Companion performs an entity recognition over the currently browsed web pages. Starting from these entities it offers the set-based browsing previously described, including the faceted filtering functionality. The entities present in the resulting set are used to perform a web-pages recommendation. Companion acts like an extra-navigational stack. The authors affirmed they created Companion in order to familiarize the users to set-based browsing by incorporating it in a classic browsing experience.

RelFinder[28] (2009) is a web application that displays the paths existing be-

[27] http://www.freebase.com/
[28] http://www.visualdataweb.org/relfinder.php

tween two entered resources in a graph form [74]. The online implementation works on several datasets including DBpedia and the Linked Movie DataBase. The path identification remains at the instances level and does not traverse the *rdf:type* properties. Indeed, in datasets like DBpedia a huge amount of instances can share the same type (e.g. *Person*). Taking into account these relations can retrieve a multitude of paths that only have minor interest. RelFinder helps the users understand the relations between the resources of interest. This task can be very cognitive and time consuming when done manually. The users can influence the graph displayed by specifying a maximum path length and activating filters. The RelFinder application identifies iteratively the paths between the entered resources of interest by length order. The authors notably mention that they allow the connections chain to change sense only once due to performance reason and because multiple changes are difficult to interpret by the users. The method uses a sequence of SPARQL queries and is agnostic to the knowledge base.

gFacet (2010) is a prototype combining graph-based visualization and faceted filtering techniques [73]. The facets are represented in the form of nodes in a graph, the arcs represent their dependencies (e.g. *birthPlace* links the facets *Person* and *Place*). One facet is passive and constitutes the results set. When the users select a filtering value in a facet it re-computes the values in the other facets and in the results set. The ability to use distant facets (facets that are indirectly related to the results set) is referred to as *hierarchical faceting* when such facets are often organized hierarchically [73]. As the effects of the interactions caused by the facets dependencies can be complex, the users can track them through colored indications. New facets can be added and they are removable. Their values can be sorted and accessed through paging and scrolling functionalities. gFacet also supports pivoting: it is possible at any moment to select a facet that becomes the new results set. The expressiveness of the gFacet interaction paradigm allows its users to build complex queries through a succession of simple interactions that does not require any technical knowledge.

Visor (2011) is a semantic browser that implements a *"multi-pivoting browsing"* paradigm [152]. Contrary to the majority of browsers Visor allows the users to start the exploration from several resources. It supports a multi-directional exploration by following several properties at a time. The prototype is built on top of DBpedia. The users first select one or several classes thanks to a keyword-search functionality. The selected classes are displayed in the form of nodes' graph using a force-based layout. Visor identifies the relations existing between the selected classes (renamed "*collections*" in the interface) and presents them in the form of arcs between the nodes. One of the key principles behind Visor is that instances data are only shown on demand. The objective is to raise the understanding of the domain explored. Thus the majority of interactions occurs at the schema level and includes adding, removing, dragging collections and asking for details about them and their relations. When the selected collections are not directly linked Visor identifies and displays on the graph the *intermediary* collections linking them. Finally the users have the possibility to create, query and export instances data

spreadsheet from the graph they built.

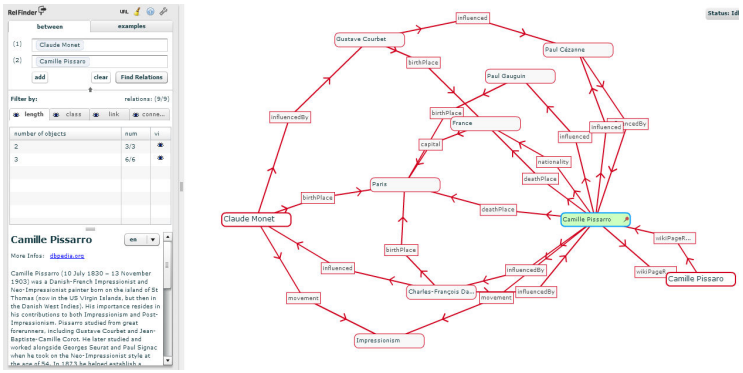

Figure 4.19: RelFinder

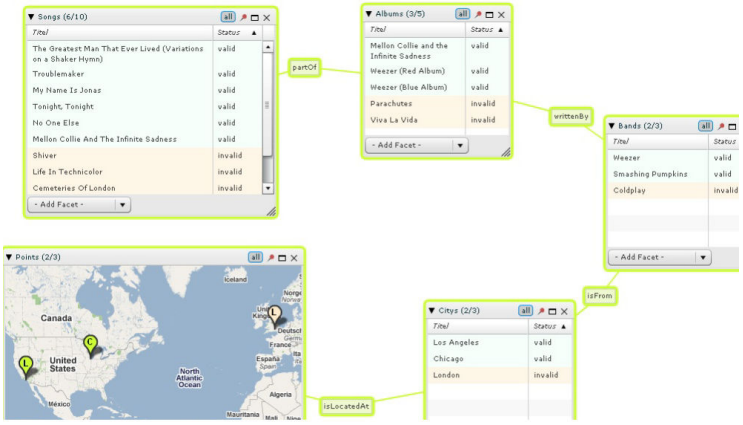

Figure 4.20: gfacet

Figure 4.21: Visor

4.3 Linked data recommenders

Several linked data exploration and discovery approaches are based on similarity or relatedness measures. Their objective is to identify the resources (*result-resources*) that are the most similar or related to an initial resource or resources' set (*seed-resource(s)*). Measuring similarities thanks to linked data datasets is a relatively new concern [173]. In the linked data context "*similarity metrics evaluate the degree of overlap between entities based on a set of pre-defined factors such as taxonomic relationships, specific characteristics of each entity, or statistical information derived from a given knowledge base*" [130]. The measures of relatedness are more structural and aim to select the most informative resources about one or several seed resources. The hypothesis is that these results-resources are strongly connected (related) in a direct or indirect manner to the seed(s). When considering linked data sources the similarity and relatedness measures are overlapping. Indeed, in order to compute the similarity, operators are applied on triples and paths that include the seed-resource(s) and result-resource(s). The triples used to compute a similarity also capture a relatedness, see figure 4.22. For example *Claude Monet* and *Édouard Manet* have both *Impressionism* stated as their artistic movement: it is a similarity insight (the two resources share a common property and value) as well as a relatedness one (they are linked through the *Impressionism* node).

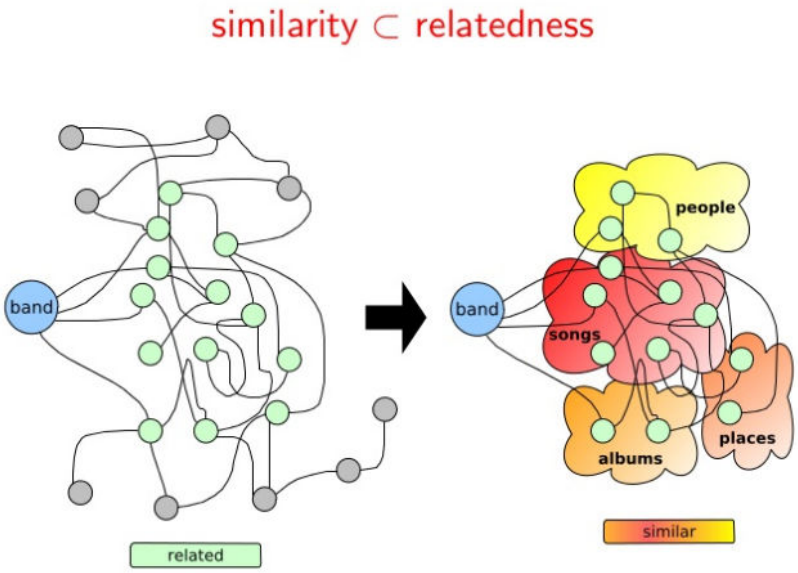

Figure 4.22: Overlap between relatedness and similarity measures in linked data datasets[29]

The metrics of similarity between words and concepts have been investigated first in the field of psychology and linguistics [183]. In the context of the semantic knowledge bases the interest about semantic similarity measure was initially motivated by ontology alignment and merging concerns. In this field, there is a

need to identify the equivalences in terms of classes and properties among different schemas. Important contributions focused on this research challenge include [44] about comparison of ontologies, [160] about determining semantic similarity among instances of different ontologies and [49] about similarity-based ontology alignment. The similarity computation between words in lexical databases was also extensively studied for various information retrieval objectives, the reader may refer to [130] for further references. One of the most used knowledge source in this context is WordNet[30]. Wordnet is a lexical database that was developed by Princeton University linguists. It indexes, classifies and organizes the semantic and lexical content of the English language. Similar initiatives in others languages also exist[31]. As rich relations are declared between the terms (e.g. hypernynmy/hyponymy, meronymy) Wordnet can support graph-based techniques in order to compute similarities between word pairs e.g. taking into consideration length or depth measure(s) for instance.

The works presented in this section propose semantic similarity metrics for linked data instances. DBpedia is the most common linked data source used due to its large coverage. Interestingly several successful approaches based on Wikipedia were proposed before the appearance of DBpedia-based ones. They use both structural and text-based metrics, see [55], [203], [205] and [59].

Similarity measures are key components of the recommender systems. Recommendation is a common and widespread feature offered by many applications to reduce the users information overload and increase their satisfaction. It is especially important for e-commerce players in order to encourage new purchases [165]. The recommendation approaches are generally classified into two broad categories:

- **Content-based filtering**: that uses items characteristics such as associated tags and structured information in order to compute the recommendations.
- **Collaborative filtering**: based on the hypothesis that similar users are interested in similar items. The objective of these systems is to find correlations among the users shared likes and dislikes for the recommendations computation. The previous interactions of a user will be used to predict the interest of others users identified as similar regarding their interest.

The majority of the linked data-based approaches presented below falls in the content-based filtering category. One important advantage of these approaches is the capacity to explain the recommendations provided i.e. presenting the users the factors of similarities and relatedness. The explanations constitute a critical aspect for the acceptance of the recommendations by the users [39]. Such explanations are not possible with collaborative-filtering approaches, notably for privacy concerns, leading to *black-boxes* recommenders. Contrary to content-based approaches the collaborative-filtering techniques also suffer from the *cold-start problem*. Indeed no

[30] http://wordnet.princeton.edu/

[31] wordnet in French: http://alpage.inria.fr/ sagot/wolf.html

recommendations related to the newly introduced items can be generated till a certain amount of users' interactions occur on them.

Recommenders and exploratory search systems are very different tools but show interesting intersections [131]. They share the common objective of assisting the users in information or resource discovery in a collection, but in very different ways. On one hand the recommenders provide direct suggestions that do not require or require minimal user interactions. On the other hand the exploratory search systems attack the discovery challenge through the angle of users' engagement and high interactivity. Moreover some linked data based exploratory search systems integrate recommendation functionalities, as we will see in the section *linked data based exploratory search systems* of this chapter. In this part we will first review linked data based recommenders that are focused on a specific type and/or domain. Second we will review more complex recommendation use-cases including cross-domain and lateral recommendations. Third we will review the semantic data recommenders recently released by the 3 major search engines.

4.3.1 Type and domain-specific recommenders

In the case of type-specific recommendations the inputs and results are constrained to a specific type of objects corresponding to a class of the ontology e.g. movies recommendations starting from a movie. These approaches are consequently domain-dependent. The cultural domains like cinema and music, where exist an important culture of recommendation, were particularly addressed.

In [147] the author presents a semantic distance measure that is used to compute recommendations: the **Linked Data Semantic Distance** (LDSD, 2010). It is based on direct and indirect paths counting between a pair of resources. The author also introduces a version with weighted paths. In this variant the dataset prevalent properties are considered more important. Six variants of the algorithm were tested thanks to a users' evaluation (counting only the direct links, only the indirect ones, combining both and using the weight function or not). The variant combining the weighted direct plus indirect links was evaluated as the most effective. The algorithm was applied on the music domain by computing the similarities between all the *Bands*[32] and *Musical Artists*[33] instances contained in DBpedia. The similarity is computed for all the pairs of targeted instances. The author stresses that the LDSD algorithm retrieves some unexpected but relevant results that can hardly be identified by a collaborative filtering technique that favors popular associations. He mentions the case of the *Tennessee Three* band which was the backing band of Johnny Cash. The algorithm was also implemented on the literature domain, see [148]. The author stresses several advantages of relying on DBpedia to compute recommendations. It includes the possibility to display the results in several languages and to use the links between the resources as explanations. The algorithm results were successfully evaluated against the recommendation

[32] http://dbpedia.org/ontology/Band
[33] http://dbpedia.org/ontology/MusicalArtist

from the LastFM musical platform. The LDSD measure was later implemented in a music discovery engine.

MORE[34] (2010) is a DBpedia-based movie recommender accessible through a Facebook application [133]. The computation is based on a semantic adaptation of the vector space model [163] called sVSM for semantic vector space model. The more features two films movies share the more similar they are. They can be linked through direct properties (e.g. *"subsequentWork"*), be the subject of triples having the same property and object (*"starring" "Robert de Niro"*) or be the objects of two RDF triples having the same property and subject. The proposed framework was implemented on the cinema domain. For the implementation the authors used an extraction of DBpedia targeting the *Film* instances. Freebase and LinkedMDB[35] triples subsets were integrated in order to provide extra-information about the films' genres. When implementing the MORE framework a human intervention is needed to specify the relevant properties related to the targeted domain. In the authors' case-study 20 properties are used for the similarity computation e.g. *starring*[36] and *director*[37]. During their evaluations the authors discovered the positive impact of taking into account the DBpedia categories on the results quality. They also noticed that taking into account of the movies super-categories (using to *skos:broader*) was decreasing the quality of the results. Indeed when going up in their hierarchy the categories become quickly too general and artificially increase the similarities. The MORE interface allows the declaration of several resources as inputs and retrieves the union of the unitary results. The users can also ask for recommendations explanations by requesting what are the shared properties between the seed(s) and the results. An interesting feature is the possibility to influence the recommendations by tuning the weights of the matrix vectors through the interface e.g. specifying that the movies' topics are more important than their directors. In other words the users can influence the recommendation ranking scheme according to their interest. A location-sensitive mobile version called Cinemappy was developed [143]. **Cinemappy** is an Android application combining the results of the MORE recommender with values representing the users' current context (social, spatial and temporal) in order to recommend movies projected in nearby movies theaters. The context values are computed according to several dimensions:

- **Social criteria**: if the users is accompanied by other person(s) it uses the nature of their relation (e.g. couple, friends, parents and children) to influence the recommendations.
- **Temporal criteria**: the fact a movie is scheduled too early for the user according to his current position discards it.
- **Spatial criteria**: Closer movie theaters are favored. The computation in-

[34] http://apps.facebook.com/new-more/
[35] LinkedMDB is a linked data base specialized about movies: http://linkedmdb.org/
[36] http://dbpedia.org/property/starring
[37] http://dbpedia.org/property/director

cludes hierarchy (the users in the same geographic area such as city district or not), clustering and co-location (e.g. restaurants near the movie theater) metrics.

Figure 4.23: MORE

[134] presents the **DBpediaRanker** (2010) which computes similarities between resources. The similarity measure is a combination of several weights including external services (search engine correlations measure with Google, Yahoo, Bing and Delicous), one link analysis measure based on the existence of a link between the considered resources (no links, a link in one direction, two links in both directions) and one textual analysis detecting if the label of the first resource appears in the DBpedia textual short description (the *abstract*) of the second one and vise-versa. The approach was implemented on an information, communication and technology DBpedia subset. This subset was extracted thanks to a dedicated module and is composed of 8596 resources. The DBpediaRanker was used in several exploratory search systems (presented in the next section).

In **Lee and al.** ([109], 2010) the authors propose a semantic association search system based on a traversal algorithm that identifies the most relevant resources starting from a keyword-query. They define two relevance metrics: a specificity measure (favoring infrequent resources and properties) and a generality one (favoring frequent resources and properties). When a user enters a query in plain-text the keywords are matched to data literals associated to the resources thanks to a frequency similarity measure. Several resources are identified as the algorithm starting points. Then the two measures are used by the traversal algorithm to produce the two rankings. The approach proposes two fixed ranking schemes (generality and specificity) and consequently two result lists. They aim to satisfy two different information needs that correspond to two perspectives i.e. get specific or general information about resource(s) of interest. The authors present several queries' results performed on a small extract of a knowledge base related to elec-

tronic appliance companies in Korea.

In **Groues and al.** ([63], 2012) the authors propose an adaptation of the Maedche and Zacharias measure [117] in order to apply it on linked data datasets. This measure takes into consideration several criterions about the considered pair of instances: a taxonomy similarity (class and category they share), a relation similarity (commons relations they share) and attributes similarity (comparison of their literal values). According to the authors the proposed measure needs to be adapted. It makes strong assumptions on the knowledge model that are not compliant with the semantic web data model e.g. each class is subsumed by only one other class. For their evaluation the authors computed the similarity between 30 movies pairs and compared the obtained value to the human judgment. They found a correlation of 0.69 between the human ratings and their semantic similarity results. According to the authors the method needs to be improved as some similarities were significantly underestimated or overestimated by the algorithm.

4.3.2 Cross-types and domains recommenders

The linked data based recommenders described in the previous sub-section target a defined type of resource and/or domain. This constraint is motivated by performance and recommendations quality concerns. It is easier to operate on a data subset as it limits the data heterogeneity and allows to process more finely the semantics (by manual declaration of relevant properties for instance). But it can also appear as a severe limitation in the linked data context. Indeed the richness of the linked data datasets and their interoperability offer an unprecedented ground for computing complex similarities and relatedness metrics. Retrieving recommendations mixing several domain of interest and/or several items classes leads to various issues [202]. It is an ongoing topic of research inside and outside the semantic web community [115]. The linked data in general and especially the multi-domains datasets like DBpedia or Freebase offer a promising ground for this research challenge.

Fernandez and al. ([51], 2011) proposed a framework to compute cross-recommendations on at least two chosen domains. More precisely it provides recommendations in a defined domain starting from instances that belong to another domain. Previous works proved the existence of latent similarities between items in the domain of music and places of interest [51] [87]. They applied the method on DBpedia data using the scenario of musical recommendations starting from tourists' attractions (e.g. "*Vienna State Opera*"). The recommendation computation is operated offline and stored in a relational database. In a first time they perform a DBpedia triples extraction of the two targeted domains. Then instances belonging to the 2 domains are linked thanks to an acyclic graph. This acyclic graph is composed of a set of classes and properties that are selected by an expert. In the case-study the 2 domains are linked thanks to 3 different semantic paths. The first one considers the city they share, the second one considers the date they have in common (same period or not). The third one is a combination of categories

and emotional tags coming from a previous work as well as lastFM music genre coming. Finally a weighted traversal algorithm is performed over this graph to generate the recommendations. The authors notice that the paths can be used as explanations. It is especially important to explain cross-domain recommendations as they can be non-obvious. The positive evaluation presented in [86] and [85] showed very encouraging results (precision over 80 percent for top 5 results). The more an artist is related through different paths to a place (city, date and category for instance) the more he is positively evaluated. The evaluation results confirm the potential of linked data and DBpedia in particular for cross-domain and cross-type recommendations. The authors mention some limits, for instance they do not use the direct relations in the actual model (e.g. *Gustav Mahler was the director of Vienna State Opera*). Such cross-domain recommendations can be highly interesting for e-commerce platforms or social networks as they contain a wide variety of items (e.g. books, films, bands).

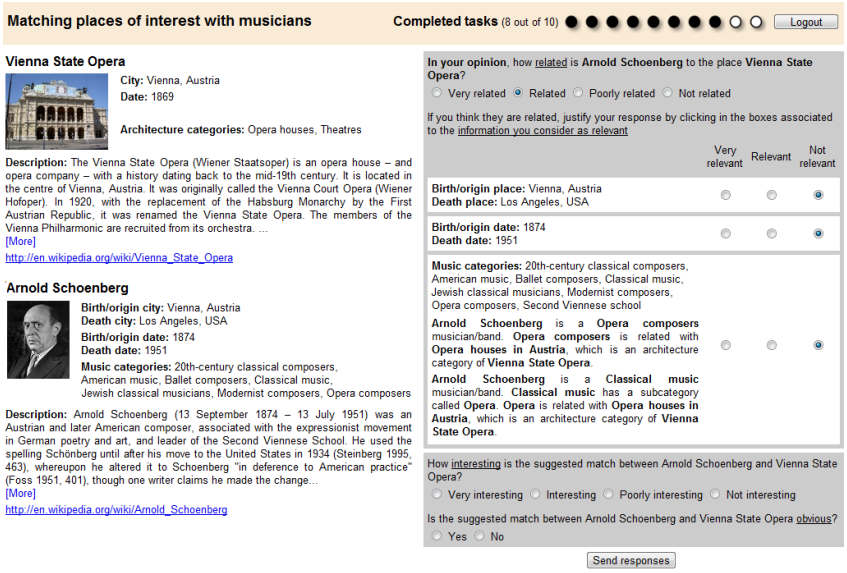

Figure 4.24: Cross-domain recommender evaluation interface presented in [51]

[174] proposed the **hyProximity** algorithm (2011). It computes lateral recommendations for open innovation purposes. A valuable approach for open innovation is to reveal topics that are lateral to a given problem. The goal of the hyProximity algorithm is to reveal hidden and unexpected associations that are relevant in the context of industrial problem solving. A set of DBpedia entities is first extracted from the targeted industrial problem description. Then the hyProximity algorithm tries to find topics (and by extension experts) that might propose an innovative solution by traversing the DBpedia graph.

Heitmann and al. ([76], 2012) and [77] introduce several challenges inherent to cross-domain recommendations and propose the **SemStim** algorithm to solve them. Contrary to Fernandez and al. the targeted domains for the recommenda-

tion are not pre-defined. The idea is that heterogeneous interests of the users can be gathered from various platforms or from a multi-domain one like Facebook in a single graph to ease recommendations. They are merged in a unique graph using DBpedia. A traversal algorithm is applied to process this graph and to compute the cross-domain recommendations. The recall and precision of the algorithm was positively evaluated against others algorithm for both single-domain recommendation (using the Movielens 100k datasets[38]) and cross-domain ones (using the Amazon SNAP dataset[39]).

4.3.3 Industrial semantic recommenders

As mentioned in the subsection 3.5.2, the 3 major search engines released between 2012 and 2013 an entity recommendation feature based on their respective knowledge graphs. The algorithms, software solutions to process the search engines knowledge graphs and run the functionalities are sensitive and mainly undisclosed, especially Google Knowledge Panel (2012) and the Bing Snapshot (2013) ones. Some technology-analysts[40] suppose that the Google Knowledge Graph is run by the Google Pregel [118] large-scale graph processing engine. Bing might be run by the Microsoft Research's Trinity graph engine [166] which was designed to handle billions of triples. If we refer to an old schema of the Trinity architecture, see Figure 4.3.3, we can observe there is a chance that Bing Satori uses RDF and SPARQL as native knowledge model and query language and not proprietary ones. On both Bing and Google interfaces the recommendations are presented under the label *"people also search for"*. This formulation suggests that a collaborative filtering technique is used, but it is only a supposition. The algorithms and approaches in general remain secret.

Yahoo gave more information about its SPARK recommender system in [20]. The Yahoo Knowledge Graph is stored in "*a native graph database*". SPARK extracts a set of features about the resources in order to compute the recommendations. It includes co-occurrence that corresponds to how frequently two entities appear together in Yahoo search query-log, Twitter and FlickR. Popularity is also computed i.e. the prevalence of an entity in a data source (Yahoo Search results, Twitter, FlickR tags). Graph-theory based features are also extracted. The final ranking is performed thanks to a stochastic gradient boosted decision tree [54]. The system was evaluated thanks to a golden-truth composed of manually evaluated results by Yahoo editors. Evaluations are run periodically to validate updates and optimization of the system. In-deployment positive usages statistics were observed in terms of numbers of click per SPARK recommendation and coverage: number of queries covered by SPARK. These two metrics are better and better since the deployment. SPARK clearly overcomes the results obtained by the previous system.

[38] http://grouplens.org/datasets/movielens/

[39] https://snap.stanford.edu/data/web-Amazon.html

[40] http://arstechnica.com/information-technology/2012/06/inside-the-architecture-of-googles-knowledge-graph-and-microsofts-satori/

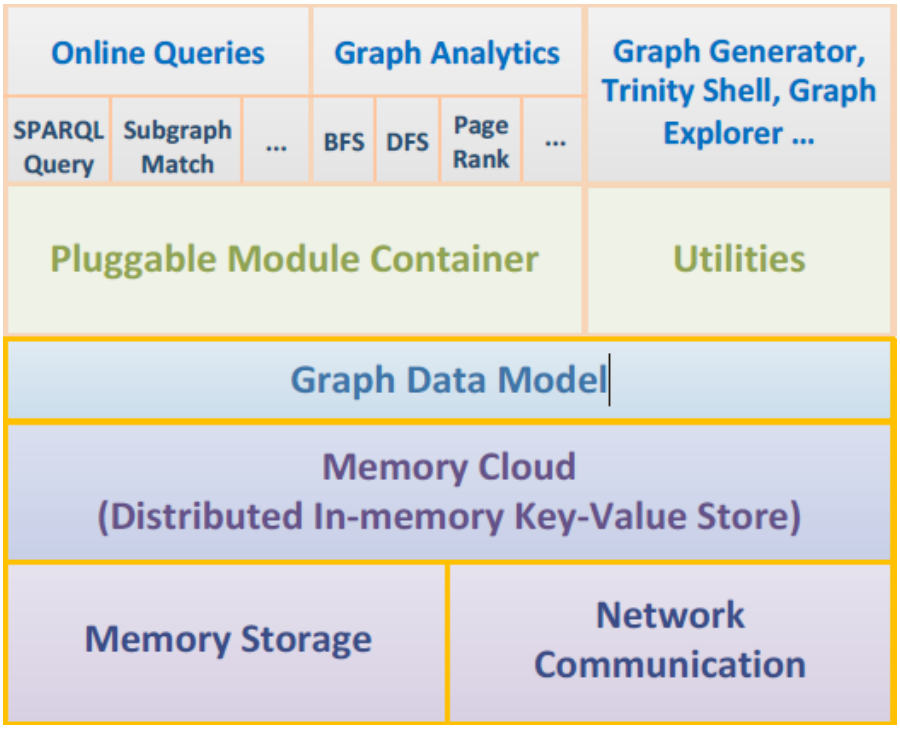

Figure 4.25: The architecture of the Trinity graph processing platform taken from [166]

The authors noticed an important trade-off between coverage and relevance. Today SPARK is deployed for English-language markets, Taiwan, Hong-Kong and Spain. The authors conclude the publication by affirming that their ultimate scope is *"question-answering, i.e. answering arbitrary keyword queries regarding entities on the web"*.

It is important to stress that the research propositions that have been applied on the linked data cloud by the semantic search community are applicable, at a formal level, on such *private* knowledge graph. However a specificity of Bing, Google and Yahoo compared to other actors it they have critical concerns about scalability and quality of service.

4.4 Linked data based exploratory search systems

At the time of writing, several exploratory search systems based on linked data already exist. Exploratory search is still at an early stage of research. They are the most advanced systems in the field of linked data based exploration. They were built in the spirit of the Gary Marchionini seminal paper [120] and often reference it. Semantic search technologies are more mature but still actively studied and strongly evolving. Consequently, using semantic search approaches to support exploratory search opens many questions today. In any case the semantic

and structured aspects of linked data makes it very promising for supporting exploratory search asks. Several research works proved that additional semantics such as tags [88] or representation of the knowledge space structure [156] have a positive influence on the exploratory search. Moreover the sense-making activity that occurs during an exploratory search task can be eased and inspired by the linked data existing structure. We can anticipate a resonance between the users' mental models (frames) and the ontological schema(s).

We identified two main approaches in the literature in terms of processing and interaction model. The first one is to allow the users to produce views on the graph through rich operators (first subsection). The second approach consists in applying algorithms that leverage the semantics in order to select and rank a small amount of data that are presented to the users (second subsection). In the second case the graph is not displayed as it is. Computed relations (such as similarity, relatedness) are shown instead. In this section we review linked data based exploratory search systems. These systems vary a lot in the functionalities they propose.

4.4.1 View-based exploratory search systems

Aemoo[41] (2012) is an exploratory search system which is based on Encyclopedic Knowledge Patterns[42] [137]. It is built on top of DBpedia. Encyclopedic Knowledge Patterns (EKP) define the typical classes used to describe entities of a certain type. For instance *"airport"*, *"aircraft"*, *"military conflict"* or *"weapon"* are part of the *"aircraft"* EKP. They specify *"the most relevant types of things that people use for describing other things"*. They were built thanks to a DBpedia graph analysis [140]. According to the authors in [140] the Encyclopedic Knowledge Patterns provide a *"cognitively-sound organization and selection of information"*. Starting from a resource of interest Aemoo presents its direct neighborhood filtered with its corresponding class EKP in the form of a graph. Aemoo consequently minimizes the cognitive load of the users by presenting them only a filtered set of nodes that are the most informative regarding the topic of interest. It is possible for the users to focus only on certain types of relations proposed by the view e.g. showing only the *Scientists* that are related to *Immanuel Kant*. Aemoo also proposes a *"curiosity"* function which presents the yopic neighborhood through an inverted EKP filtering. This function aims to reveal unexpected knowledge. Aemoo also analyses Twitter and Google News feeds using named entity recognition using the resource of interest's label. If it detects an entity in these feeds it adds it to the results e.g. a news article that mention both *Immanuel Kant* and a *Scientist*. Consequently relations that do not exist in DBpedia can be dynamically exposed in Aemoo. The application offers a breadcrumb that allows the user to go back and forth. Aemoo also provides explanations by showing the cross-references between the seed and the result in their Wikipedia pages, Twitter and Google News (depending on the link provenance).

[41] http://wit.istc.cnr.it/aemoo
[42] http://www.ontologydesignpatterns.org/ekp/

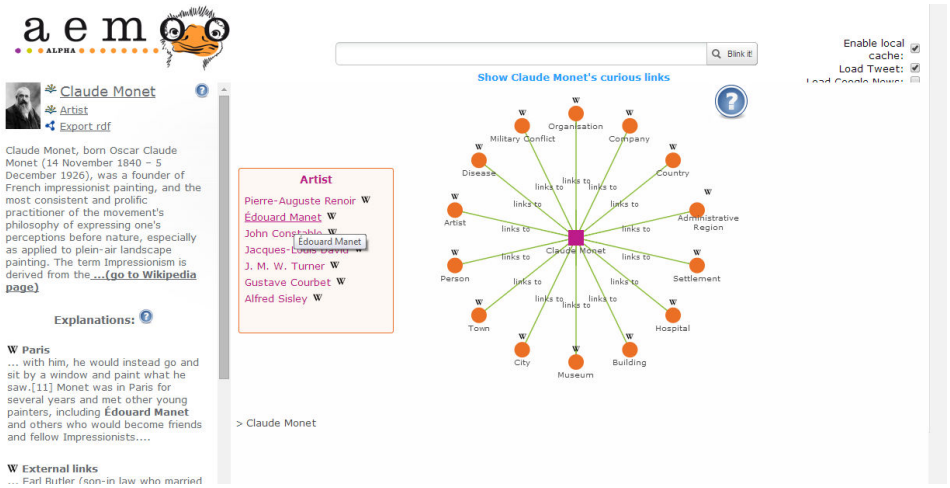

Figure 4.26: Aemoo

Linked Jazz[43] (2013) is a project aiming to reveal the network of social and professional relations within the American jazz community [149]. The objective is to capture such relations from transcripts of jazz people interviews available in museums, galleries, libraries and to publish it in RDF following the LOD principles. The jazz community is an interesting application case as it is a dense and complex network that is well documented.

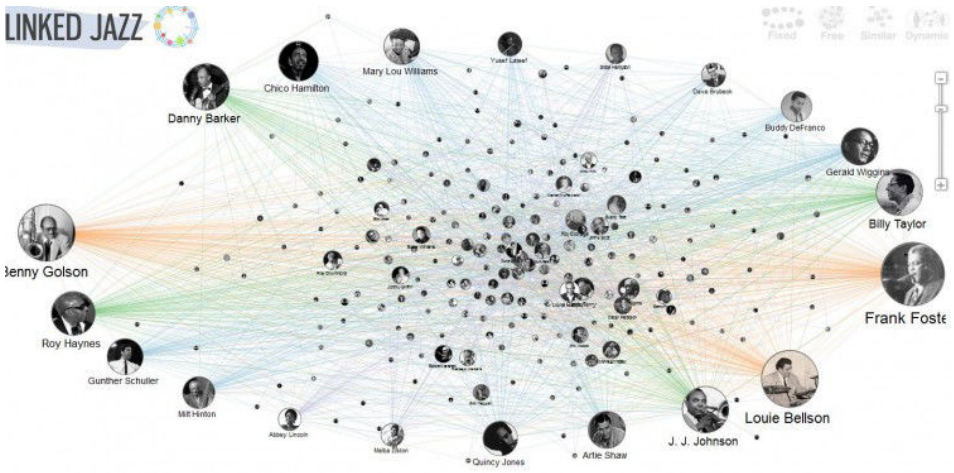

Figure 4.27: The Linked Jazz network visualization web application

The method used to reach this goal is a combination of several steps of automated processing and crowd-sourced curation. They are listed hereafter:

- Building the *Linked Jazz Name Directory*: first a DBpedia extraction is performed. Then this extraction is completed and cleaned thanks to the library

[43] http://linkedjazz.org/

of congress catalog[44] and the Virtual International Authority File[45] data e.g. adding nicknames to the directory. The name, birth and death date are compared and a degree of confidence is assigned regarding the percentage of positive matches.

- Cleaning the names in the directory: the identified names and their variants sorted by degree of confidence are accessible online and can be approved, corrected or deleted by anybody through the *"name mapping and curator"* web application[46].
- Processing of the transcripts: the names present in the Linked Jazz directory are identified in the transcripts thanks to the Natural Language toolkit 2.0 platform[47]. Others resources such as the places, albums, songs, businesses, and partial names are also identified. The transcript analyzer also structures the text by clearly separating the questions and answers for the following human analysis step.
- Identifying the nature of the relations in the transcripts by using a crowdsourced approach: it is possible to easily explore the transcripts and to manually identify the nature of the relations they contain through a web application[48]. The available types of relation are predefined and include for instance *acquaintance of, close friend of, played together, was mentor of*.

The dataset obtained can be explored through a visualization tool[49]. It gives an overview of the jazz community network and supports more focused explorations e.g. individual connections, shared connections, dynamic network creation by manual selections. All the tools developed in the context of this project are open-source and domain-agnostic. They can be reused for other applications. Dumps of the data are also available online.

4.4.2 Algorithm-based exploratory search systems

The first system that makes use of the DBpediaRanker, presented in the previous section is the **Semantic Wonder Cloud** (SWOC, 2010) [135]. SWOC was inspired by the Google Wonder Wheel[50] and helps the users discover new knowledge around topic of interest. It uses the ICT domain resources' similarities computed by the DBpediaRanker. The users first select a topic of interest thanks to a DBpedia lookup. Then SWOC presents a graph-view of this topic surrounded by its 10 most similar concepts. In other words the application presents only a subset of the most informative related topics to the user. The connections that are

[44] http://catalog.loc.gov/
[45] http://viaf.org
[46] http://linkedjazz.org/tools/name-mapping-tool-and-curator/
[47] http://www.nltk.org/
[48] http://linkedjazz.org/52ndStreet/
[49] http://linkedjazz.org/network/
[50] http://www.googlewonderwheel.com/

shown are not necessarily direct neighbors in DBpedia graph. Consequently distant nodes in the DBpedia graph can be presented as neighbors in SWOC. It eases the process of exploration and unveils hidden knowledge through the exposition of associations that does not exist in the original graph. SWOC keeps track of the resources that have been shown to the users during the exploration so he can move backward to the previous steps of exploration.

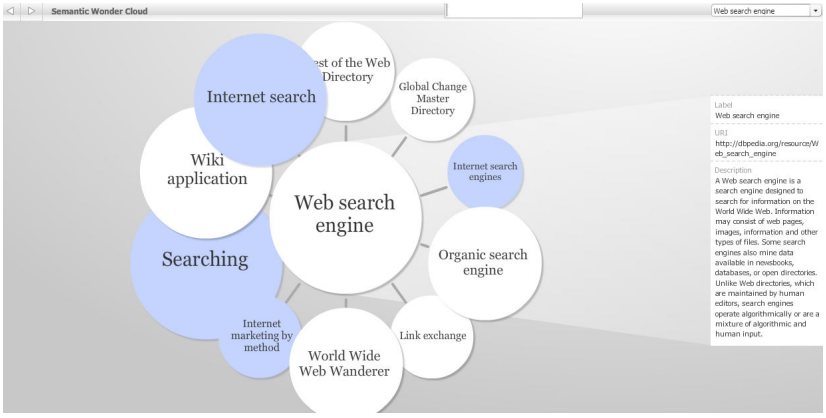

Figure 4.28: Semantic WOnder Cloud

The second system that retrieves the DBpediaRanker similarities is **Lookup Explore Discover** (LED, 2010) [132]. LED is an exploratory search system that proposes query-resources suggestions that are semantically computed and presented in the form of a tag cloud. The user first enters a topic of interest using a DBpedia lookup. Then semantically similar DBpedia tags are proposed. Based on the query composed, the system retrieves an aggregation of results coming from several major search engines (Google, Yahoo, Bing) as well as a news feeds (Google News) and a microblogging service (Twitter). Thanks to the tags the users can refine their queries and explore more easily an unknown domain. A user evaluation showing positive results is presented in [134].

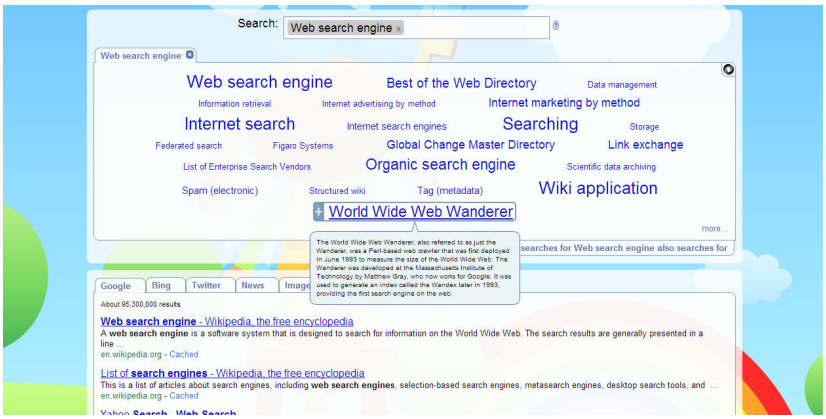

Figure 4.29: Lookup Explore Discover

The Linked Data Semantic Distance, presented in the previous section, is now implemented by the **Seevl**[51] musical discovery web platform (2012). A plugin is also available for the music platform Deezer[52] and Youtube. Seevl uses the DBrec algorithm to retrieve artist recommendations, it is also possible to use the properties-values associated to an artist to discover other artists e.g. its musical genres, collaborations, record labels, band members. Seevl also provides triple-based explanations of the recommendations e.g. a band is recommended because its genre and place of origin are the same as the query ones. The music is directly playable on Seevl (fetched from Youtube). Seevl is a domain-dependent exploratory search system specialized on the musical domain.

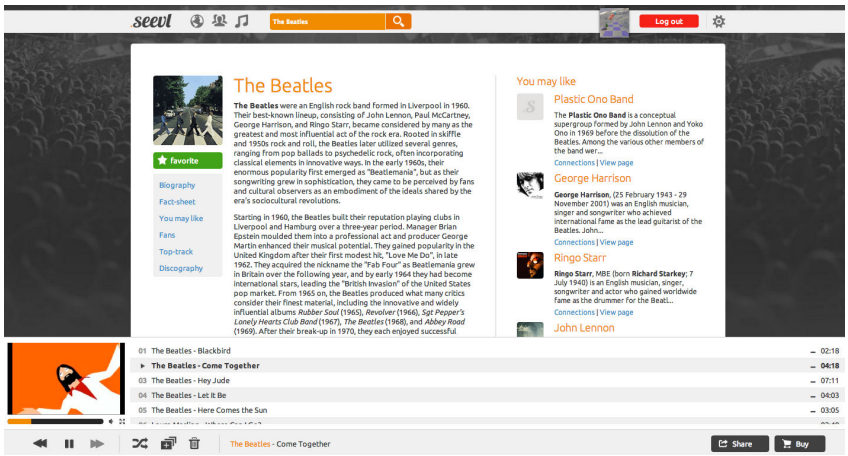

Figure 4.30: Seevl

[191] presents **Yovisto**[53] (2010) which is an academic videos platform hosting lecture recordings, conference talks, etc. Yovisto offers a *"fine granular based video index"* allowing its users to navigate in the content. The videos are collaboratively annotated by the users and benefit from automatic metadata generation. The system proposes a search functionality. It also provides a linked data based exploratory feature in the form of an entity suggestion related to the current query. According to the authors "*broadening the scope and suggesting nearby related search alternatives is the tasks of an exploratory search feature*". The objective is to propose *"guided route"* to the users in order to help them to explore the information space and to favor discoveries. It is especially useful at the beginning of an exploration when they lack salient keywords. The users can incrementally solve their information needs by performing sequences of suggested queries. These suggestions are computed over DBpedia thanks to a property ranking based on a combination of heuristics. They make explicit relation that can not be directly derived from the Yovisto metadata. The heuristics are based on structural and statistical

[51] http://play.seevl.fm/
[52] http://www.deezer.com
[53] http://www.yovisto.com/

graph characteristics. They take into account various criteria such as the connection strength (e.g. unidirectional outgoing, unidirectional ingoing, bidirectional), the properties-frequency and some schema-based measures. Yovisto also proposes an in-session breadcrumb functionality that helps the users to come back to a previous search state. The authors validated their approach thanks to two rounds of experimentations. They stress that human evaluation is mandatory when considering exploratory search systems. First they received the help of 72 users to build a golden-truth. It is composed of 115 entities for which the users specified the most important facts and associations in their opinion, with the help of the Wikipedia page. At the end the authors obtained a set of 2.372 distinct resources assignments (total of 5.225). Using them as a golden truth they evaluated a large number of heuristics combinations in order to observe the variations in terms of precision and recall. Finally they performed a qualitative evaluation using 9 different task-based scenarios. A set of 11 out of 19 users performed the tasks with the exploratory search feature enabled whereas 8 users used Yovisto without the exploratory search feature. The comparison between the results of the 2 groups showed the benefits of the exploratory search feature: more search tasks were accomplished, the user satisfaction and the system helpfulness were judged better (1.82 vs 1.11 on 4 and 2.29 vs 1.66 on 4). It brought significant improvements with only a small loss in perceived familiarity (0.97 vs 1.06 on 4). The processing time per task and the number of results found were also better.

Figure 4.31: Yovisto

inWalk[54] is a web application designed for high-level linked data exploration

[54]http://islab.di.unimi.it/inwalk

[28]. It allows the users to interact with a graph of clusters named *inCloud*. An inCloud is a set of linked data based clusters, computed according to a manually declared list of properties. The clusters are related to each others by a semantic proximity link. The current inWalk web application is implemented on a Freebase subset related to athletes and celebrities. Each cluster is constituted of a set of resources and is labeled with *the top-k most representative labels/types* of the cluster's resources (e.g. "*american/film*", "*italy/olympics*"). The semantic proximity is used for the display i.e. closer clusters in term of semantic proximity appear closer in the graph. InWalk aims to give an overview on the explored domain and to ease its appropriation by the users. It shifts the general granularity of a single resource to semantically aggregated sets. inWalk proposes a proprietary query language to manipulate the data as well as a search with auto-completion that restrains the inCloud to the elements that match the query. Details of the computation (clusters, labels, similarity) are available in [27].

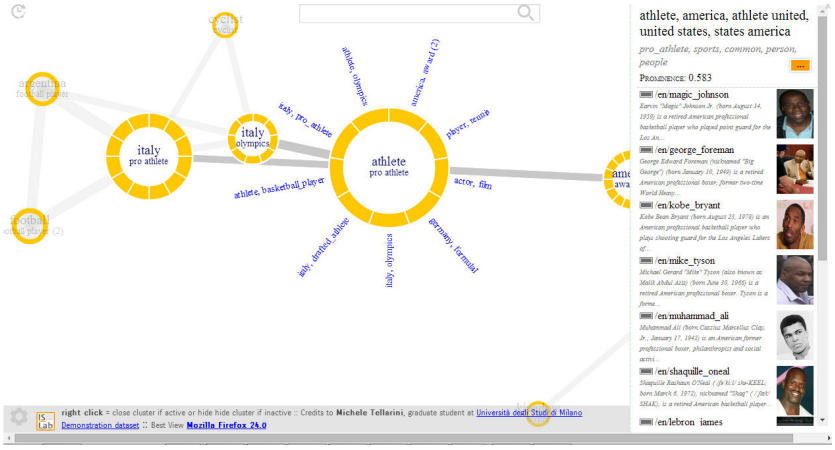

Figure 4.32: inWalk

4.5 Discussion

The Figure 4.33 graphically shows the evolution of the research over the time. It appears that during the first development phase of the semantic web (2001 - 2007) a wide range of browsing paradigms were investigated. Simple systems inspired by the classic web pages browsing experience were conceived. In such systems the data is often displayed in rows of properties and objects/values. The users browse from one resource to another using the hyperlinks. Simple semantic-based support was sometimes available (data provenance indicator in the Marble browser[55] for instance) but the semantics were not already leveraged to build disruptive browsing paradigms. During this period many visual and faceted browsers were also conceived. A possible reason of the success of such approaches is that the systems often operated at the time on small (and sometimes curated) datasets. The

[55]http://wiki.dbpedia.org/Marbles

data size and homogeneity was favorable to both visualization and faceted interfaces. The first linked data browsers constituted a case of *"diligence effect"*[56]: at the beginning of a technical invention, new tools are created using old protocols.

The Linked Open Data initiative (2007) renewed the research in the field of semantic browsing and search. The quality, size and coverage of generic datasets like DBpedia opened the door to the elaboration of more complex approaches. The innovations concerned both the data processing and the interaction models. Innovative browsing paradigms were investigated including in particular set-based (Parallax [81]), multi-pivoting (Visor [152]) and hierarchical faceting (gFacet [75]) ones. Such systems allow the users to produce sophisticated views on data through rich operators in order to ease their understanding and exploration.

Thanks to the LOD datasets computing linked data based recommendations became also possible (2010) and showed very encouraging results. The computation was domain/type constrained at the beginning (MORE, LDSD [148], DBpediaRanker [134], Lee and al. [109]) but more complex approaches including cross-domain (Fernandez and al. [51]) and lateral recommendations (hyProximity [173], SemStim [77]) were later explored with success. Recently (2012 - 2013) the 3 major search engines deployed their knowledge graph based entity recommenders. Such deployments on mainstream services, subject to critical quality issues, confirm the maturity of linked data based recommendation techniques.

Nearly at the same time linked based exploratory search systems appeared. Some of the similarity measures implemented by the recommenders constituted a basis for such systems. The DBpediaRanker was integrated in SWOC [135] and LED [132] (2010). The LDSD measure was implemented by the Seevl platform (2012) to compute its recommendations. Other exploratory search systems based on various forms of processing also appeared including heuristic-based ranking (Yovisto [189]) semantic pattern-filtering (Aemoo [137]) and clustering (InWalk [28]). The systems published till today are very heterogeneous in terms of functionalities.

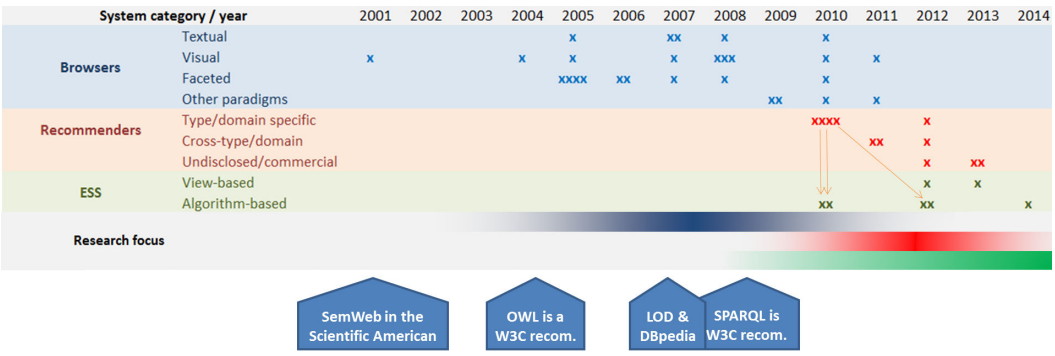

Figure 4.33: Time-line of exploration and discovery systems based on linked data

[56]https://dictionnaires.ensad.fr/cairn/resume.php?ID_ARTICLE=MATE_082_0089

The Figure 4.35 gives an overview of the most advanced systems for exploration and discovery using linked data. It includes the innovative semantic browsers, the recommender systems and the exploratory search systems. As the human engagement during the exploration tasks is critical we only included the systems having a user interface in this summary. We comment the table below by focusing on important interfaces and semantic search criteria. For each aspect we describe the limits and the achievements observable in the literature. We also mention the opportunities we see to overtake the actual published approaches. It is noticeable that the strengths and the weaknesses of the systems vary a lot. First, it is due to their heterogeneity. Second, the researchers and developers often concentrate their effort on a specific functionality or on a limited set of functionalities within the system. Third they tend to propose contributions where nothing was done before.

4.5.1 Human-computer interaction aspects

In this part we review the human-computer interaction aspects of the systems. To structure the analysis we remind below the diagram we proposed in chapter 2, see Figure 4.34. We enumerate the desired effects of an exploratory search system and present the principal limits and achievements encountered in the literature as well as the opportunities we see:

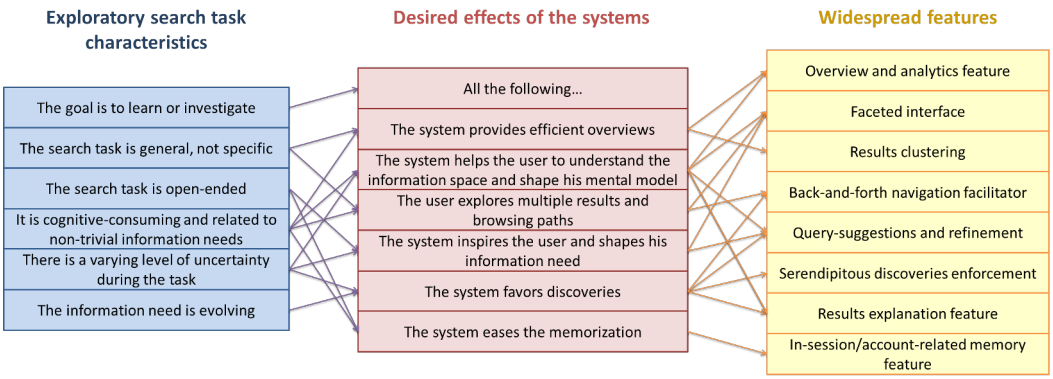

Figure 4.34: Reminder - relation between exploratory search tasks characteristics and widespread systems features

The system provides efficient overviews: over-viewing the results space is important to understand the data, especially when exploring unknown or poorly known domains. According to [61] a good overview "*provides users with an immediate appreciation for the size and extent of the collection of objects the overview represents, how objects in the collection relate to each other, and importantly, what kind of objects are not in the collection*". Overviews can deeply assist an exploratory search tasks at its beginning when the users have no knowledge about the explored domain. It is a factor of guidance in a context of high incertitude (see chapter 2, section 3).

System	Parallax	Relfinder	gFacet	Yovisto	MORE	SWOC	LED	TasteKid	Visor	Aemoo	Google K. Panel	Yahoo SPARK	Bing Snapshot	Seevl	inWalk	Linked jazz	SimilarKind
Type	Browser	Browser	Browser	ESS	Recommender	ESS	ESS	Discovery engine	Browser	ESS	Recom-mender	Recom-mender	Recom-mender	ESS	ESS	ESS	Discovery engine
Sub-type	Set-based	Paths visualization	Hierarchical & faceted	Heuristic based query recos.	Algo, type constrained	Algo, domain constrained	Algo, domain constrained	Proprietary	Multi-pivoting	Semantic pattern views	Undisclosed	Content-based	Undisclosed	Algo, type constrained	Clustered-based	View-based	Undisclosed
Year	2009	2009	2010	2010	2010	2010	2010	2010	2011	2012	2012	2012	2012	2013	2013	2013	2013
Provenance	Research	Research	Research	Research	Research	Research	Research	Industry	Research	Research	Industry	Industry	Industry	Research	Research	Research	Industry
IR	Parallax	Relfinder	gFacet	Yovisto	MORE	SWOC	LED	TasteKid	Visor	Aemoo	Google K. Panel	Yahoo SPARK	Bing Snapshot	Seevl	inWalk	Linked jazz	SimilarKind
Multiple seeds, method	Set browsing	Instances paths	Classes dependencies	No	Yes, sum	No	Yes, sum	Yes, undisclosed	Classes paths	No	No	No	No	No	No	Instances paths	No
Matching	String-match	Direct match (lookup)	Direct match (lookup)	String-match	Direct match (lookup)	Direct match (lookup)	String-match	Direct match (lookup)	Selection	Direct match (lookup)	Entity recognition	Entity recognition	Entity recognition	Direct match (lookup)	Direct match (lookup)	Selection	Direct match (lookup)
Implemntation data	Freebase subset	DBpedia, Linked MDB, LOD,	Dbpedia	DBpedias	DBpedia subset	SWOC	DBpedia subset	Freebase subset	DBpedia	DBpedia	Google KG	Yahoo KG	Bing Satori	Dbpedia subset	Freebase subset	Linked jazz KB	Freebase subset
Main algorithm	None	Sequence of SPARQL queries	None	Set of heuristics	sSVM	DBpedia Ranker	DBpedia Ranker	Undisclosed	None	EKP-based filtering	Undisclosed	Stochastic gradient decision tree	Undisclosed	Dbrec	Clustering	None	Undisclosed
HCI	Parallax	Relfinder	gFacet	Yovisto	MORE	SWOC	LED	TasteKid	Visor	Aemoo	Google K. Panel	Yahoo SPARK	Bing Snapshot	Seevl	inWalk	Linked jazz	SimilarKind
Starting point	Keyword search	Multiple lookups	Multiple lookups	Keyword search	Lookup(s)	Lookup(s)	Keyword search	Lookup	Facets selection	Lookup	Keyword search	Keyword search	Keyword search	Lookup	Lookup	Manual selection	Lookup
Principal layout	List	Graph	Graph of facets	List	List	Graph	Tag cloud	List	Graph of facets	Graph	List	List	List	List	Graph	Graph	List
Additional visualizations	Maps, timelines	No	Graph	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Query expansion	No	No	Yes	Yes	No	No	Yes	No	No	No	No	No	No	No	No	No	No
Perspective-settings	N/A	N/A	No	No	Vectors-based (n)	No	No	No	N/A	2 filtered views	No	No	No	No	No	No	No
Browsing	Set-based	N/A	No	One-to-one	No	One-to-one	One-to-one	One-to-one	Facets pivoting	No	One-to-one	One-to-one	One-to-one	No	Clusters-based	One-to-one	One-to-one
Facets, filters use	Yes	Yes	No	Yes	No	No	No	No	Yes	No	No	No	No	Yes	No	No	Yes
Results explanations	N/A	N/A	Facets dependencies	No	Common properties	No	No	No	N/A	Wikip. cross-references	No	No	No	Common properties	No	Textual	No
Memory-features	In-session history	No	No	In-session history	No	In-session history	No	No	No	In-session history	No	No	No	No	No	No	No

Figure 4.35: Summary of the existing linked data based discovery and exploration tools.

- **Limits**: an obvious limit of early textual browsers is that they only show the direct neighborhood of the browsed node. The visualization approaches gave better results in terms of over-viewing but they were often applied on manually-crafted datasets having a limited size. Such approaches are often optimized to satisfy specific use-cases and do not scale in the linked data context where the datasets are very large and heterogeneous. Some systems also limit the number of results they retrieve and deprive the users from using peripheral information in order to rise their understanding of the domain explored e.g. SWOC [135] retrieves the 10 most similar results only.
- **Achievements**: several applications are specifically built to favor the understanding of the data space by adopting a *schema first, instances second* design (Visor [152] and gFacet [75]). The users interact first with the ontological schema and have access to the instance data in the second time. This approach is powerful for linked data experts but also introduces a high risk to confuse the lay users that are not familiar with this level of abstraction.
- **Opportunities**: we see an opportunity in graphically presenting a high variety and amount of results when a topic is explored. It can raise the users' domain understanding by allowing them to rely on peripheral information. Exposing simple schema information together with the instances can also have a positive effect by adding elements of structure without confusing the users.

The system helps the user to understand the information space and shape his mental model: The execution of exploratory search tasks provokes an intense sense-making activity. The systems have to help the users in the identification of salient informational aspects to help them to deepen their understanding and to guide them in the search process. In the particular case of linked data based systems there is an interesting potential resonance between the ontological schema and the users' mental frames (as defined in the sense-making theory).

- **Limits**: the lack of structure of some systems' results space is a problem: flat lists of heterogeneous resources are presented without any explicit logic. Several systems constitute black-boxes and retrieve unexplained results, without giving explanations about them. The linked data graphs should be exploited to offer more transparency to the users.
- **Achievements**: six systems make use of facets which constitute a powerful sense-making factor by explicitly unveiling the important dimensions of the informational domain explored. Explanation features are also mandatory to understand the relations between the resources exposed and to reveal important elements of context. 5 systems offer result explanation features.
- **Opportunities**: it is critical to expose salient dimensions of the informational domain. Faceted interfaces are efficient for this purpose. At the same time we see an opportunity in offering a plurality of explanations about the results.

Offering multiple explanations can help several types of users to shape their mental frames. Moreover they will reveal different elements of context, in different forms. Putting information in context is critical during exploratory search.

The users explore multiple results and browsing paths: the cost to be engaged in a browsing path has to be low and perceived as low:

- **Limits**: the interaction models underlying exploratory search are complex. And the sequences of interactions can be difficult to remind. Several systems do not offer any function to come back to a previous state of exploration. The users have to remind the sequences of interactions and perceive the cost of being engaged in a browsing path as high. They consequently explore less and potentially miss interesting discoveries.
- **Achievements**: 6 systems propose in-session memory feature, also known as breadcrumbs. The Parallax [81] functionality goes further by allowing the editions of previous interactions e.g. deactivating a filter. Such editions lead to the re-computation of the subsequent sets of results. The breadcrumb navigation features considerably lower the perceived cost of browsing by freeing the user memory from interactions sequences tracking. Other interaction models are flexible and allow to easily modify the informational configuration to reveal new aspects of the topic explored. For instance gFacet [75] and Visor [152] allow to dynamically modify the results space structure through simple interactions: adding or removing facets for instance.
- **Opportunities**: Apart from the use of breadcrumbs we see an opportunity in dynamic interface proposing progressive layers of information and allowing to easily come back to a previous exploration step.

The systems inspires the user and shapes his information need: during exploratory search the information need is progressively shaped through data exposure. However the systems can actively assist the users during their query formulation. Ideally they should support the users in the expression and the understanding of the query:

- **Limits**: it is noticeable that the vast majority of systems implying a query or a selection do not help the users in this task. This problem is especially hard in the context of semantic search where the input is often in the form of seed-resources. It gives far less possibility than in the case of keyword-queries where multiple combinations of keywords can be suggested. The selection and understanding of the inputs issue is a bit mitigated by some lookup search bars which show important resources' information e.g. type and abstract in Aemoo [137].
- **Achievements:** two systems are particularly interesting regarding the assistance in the query composition precisely because it is their main contribution:

Yovisto [189] and Lookup Explore Discover [132]. They both rank the related query topics; using respectively heuristics and the DBpediaRanker [134].

- **Opportunities:** We understand the problem of the query composition assistance as a problem of lack of input expressiveness. To overtake the actual limitation we need to propose a query model that is more flexible, expressive and that can open the door to finer assistance in the composition. Being able to only query with *monolithic* instances prevent from new possibilities in terms of information need representation and related assistance.

The system favors discoveries: by nature all the systems supporting linked data exploration favor discoveries at some point. However, the systems can go further through processings that aim to specifically uncover unexpected information about the topic explored.

- **Limits**: the main limit we see is that the majority of algorithmic-based systems retrieve the most obvious results without giving alternatives to their users. For instance the recommender systems propose recommendations that have a very high similarity with the query. Some users might be interested in systems that retrieve less obvious results e.g. for rising their peripheral knowledge about a topic.
- **Achievements:** by nature all the tools listed in this review favor discoveries at some point. Especially because the algorithms employed are content-based and susceptible to expose unexpected piece of information e.g. the link between the Tennessee Tree band and Johnny Cash in Seevl. Such information would be probably masked by collaborative filtering approaches. One system goes further by explicitly showing curious information about the topic explored. Indeed Aemoo [137] proposes a *curiosity* functionality which expose topics curious links by identifying unusual links at the schema level (using EKPs [140]).
- **Opportunities:** We see an opportunity in proposing flexible algorithms that modify their behavior to match a desired level of surprise/unexpectedness. To go further we can imagine systems that adapt their processing regarding the identified level of expertise of their users.

The system eases memorization: during exploratory search the users seek to rise their knowledge about a topic. It implies a consequent effort of memorization that needs to be assisted.

- **Limits**: The quasi-totality of the systems do not help their users to memorize the results they discovered. None of these systems propose account-related collections to save the discovered results along the successive search sessions.
- **Achievements:** In-session memory features avoid the users to memorize their sequences of browsing. However such functionalities are session-dependent and do not persist over time whereas the exploratory search tasks

often occur through several sessions distributed over the time. Visor [152] addresses this need of persistence by offering an export functionality. Nevertheless it requires a complex manipulation that is not accessible to all the users.

- **Opportunities:** There is a need to develop account-related memory features that support knowledge acquisition and accretion for long-lasting interest. Collaborative approaches which gather several points of view and advanced knowledge organization (e.g. mind maps) are relevant axes of development for such functionalities.

4.5.2 Semantic search aspects

In this part we review the semantic data processing side of the systems. We focus especially on two aspects that are critical and that vary a lot in the literature: the data and the processing.

Data: the data constitutes the raw material for all the systems. As mentioned previously the LOD datasets made possible new forms of processing that are implemented by the most advanced systems of our review.

- **Limits**: the obvious limit we see is that the data source used by the systems is fixed. They operate on top of one (sometimes gathered from several sources) dataset for all the queries. Moreover, some approaches operate on limited dataset subsets and necessitate a manual intervention. These approaches discard many triples that could enhance the results by adding elements of context.
- **Achievements**: of course the main achievement in the field is that *real world datasets* (and not manually crafted ones) were used with success to source the systems. In this context Freebase and especially DBpedia were extensively used by the research community. It is noticeable that the Linked Jazz [149] RDF knowledge base creation was motivated by an exploration objective. The knowledge graphs built by the major search engines may reach a new level in terms of coverage and quality. Unfortunately their public access is not possible at the moment and not planned.
- **Opportunities**: the LOD is distributed by nature and it is constantly evolving. We see an opportunity in creating a system that is able to select the knowledge source to be processed regarding the query. Along with this idea we can also imagine a system that query and merge data coming from several sources in order to compute richer results.

Processing: the data processing is one of the key component of the exploratory search systems together with the interaction model. Several techniques were investigated, they differ a lot from one system to another.

- **Limits**: the algorithms implemented by the systems have diverse objectives but the vast majority share the following limit: they propose a unique results selection and ranking scheme. In other words, there is only one result set associated to a query. This can be considered as a severe limitation in the context of exploratory search where the knowledge of the users evolve at the rhythm of the search sessions. They might be interested in modulating the retrieved results according to a variety of criteria including their expertise. Moreover the objects (topics) described in linked datasets are often rich, complex and could be approached in many manners. It is also noticeable that the data freshness issue is not mentioned in the existing literature.
- **Achievements**: generally the algorithms proposed were positively evaluated. Several initiatives aimed to benefit from the domain traversal characteristic of datasets like DBpedia by proposing advanced computing. They include lateral and cross-domain recommendations. The evaluations of such approaches were also positive. Two systems offer a flexible processing that aims to unveil a plurality of perspectives about the topics of interest. The first one is the MORE movie recommender [133]. The users can tune the importance of several vectors (e.g. director, music, starring) in order to influence its recommendations. The second one is Ameoo [137] that proposes to unveil topics curiosities.
- **Opportunities**: we see an opportunity in developing a framework that overtakes the actual approaches by being able to support a wide range of different explorations (corresponding to different use-cases) starting from a topic of interest. It notably includes the possibility to generate various topics perspectives regarding diverse facets of interest, levels of expertise and cultural prisms. We also want the users to be able to combine several interests in a single query to explore resources of interest at their crossroad. This case is currently not supported by the algorithmic-based exploratory search systems. Finally the system has to leverage the distributed aspect of the LOD. It also has to address the data freshness issue as the LOD datasets evolve over the time.

4.6 Conclusion

This chapter closes the state-of-the-art review of this thesis. In chapter 2 we introduced exploratory search and focused more particularly its positioning in the field of search, its definition, systems and evaluation. Exploratory search is poorly solved by actual popular search solutions and is an active research topic. In chapter 3 we introduced semantic search, which refers to the incorporation of semantics in search approaches and systems. We focused more particularly on the actual phenomenon of structured data publication, the semantic web and linked data visions and realizations, the semantic search systems' main concepts and deployments.

Semantics offer possibilities to solve complex search queries and needs, including exploratory search ones.

Chapter 4 is at the crossroad of exploratory search and semantic search. It presents the systems that support the exploration and discovery of linked data. We described a variety of linked data browsers, recommenders and exploratory search systems by focusing more particularly on the human-computer interactions and semantic search aspects. We analyzed the evolution of the research and identified the main limits and achievements encountered in the literature. We also presented the opportunities we see to overtake the actual approaches and to build a successful linked data based exploratory search system. Starting from them we propose in chapters 5 to 8 a framework and a web application for exploration and discovery over DBpedia: Discovery Hub.

CHAPTER 5

Relevant resource selection by semantic spreading activation

Contents

5.1 Introduction

This introduction opens the contribution part of the thesis. Relying on the previously identified opportunities we propose an algorithm for linked data based exploratory search. In this chapter we detail a novel algorithm and its variants. It is designed to leverage the linked data semantic richness in order to automatically select and rank a set of resources that are informative about a topic of interest. Its implementation and evaluation are described in the upcoming chapters.

We chose to develop an algorithm-based exploratory search approach, as defined in chapter 4. Such approaches retrieve a set of algorithmically computed results that are then explored by the users through an interface. In our opinion

the view-based approaches are relevant for expert users only as they often propose complex interaction models. The algorithm-based approaches are more automated and intuitive, but often at the cost of the interaction precision. To mitigate such tradeoff we will propose in this chapter several variants of our algorithm to support a more flexible exploration.

The linked datasets semantic richness offers plenty of possibilities to sort, prioritize, discard, rank the resources by using the knowledge they contain at the instance and schema levels. At the same time their volume and heterogeneity constitute difficulties for designing algorithms that make sense of data. To lower this difficulty we based our processing on an algorithm that is well-known, documented and that has proven its efficiency for information retrieval: spreading activation. Moreover several research works have shown in the past the value of leveraging the semantics to enhance spreading activation. We propose a spreading activation algorithm adaptation that leverages the graph richness thanks to semantic filtering and similarity. The double objective of this semantic sensitiveness is to increase the algorithm relevance and lower its cost of execution.

In this chapter we will review (5a) the spreading activation basis of the algorithm: its origins, its core formula, its applications for information retrieval, (5b) we motivate our choice in favor of spreading activation, (5c) we present the formalization of our semantic adaptation for a query composed of a unique resource (referred to as monocentric), (5d) we extend the formalization of the version for a *composite* query having several resources as inputs (referred to as polycentric), (5e) we introduce the advanced querying functions based on the algorithm, (5f) we discuss the design of the algorithm. All the formalizations are illustrated by simple and fictive examples, freely inspired from DBpedia. The formalization of the monocentric, polycentric and advanced querying algorithm variants were respectively published in [128], [123] and [126].

5.2 Spreading activation basis

5.2.1 Origins

Spreading activation has its roots in cognitive psychology, more particularly in the study of human memory phenomena and operations. The spreading activation theory captures both the way the knowledge is stored in the memory (semantic network) and the way it is processed (spreading activation model). In 1966 Quillian modeled for the first time the memory in the form of a semantic network which is "*an expression of knowledge in term of concepts, their properties, and the hierarchical class relation between the concepts*" [157]. In 1969 Collins and Quillian asserted that, in the memory, information are stored in categories which are each other logically and hierarchically related [32]. For instance on Figure 5.1 *bird* is included in the broader category *animal* and is divided into narrower categories such as *canary* or *ostrich*. They also introduced the cognitive economy phenomenon: high level information such as *animal has skin* are applied to subdivisions such as *canary has*

skin. The information is stored once at the highest level and is not unnecessarily repeated.

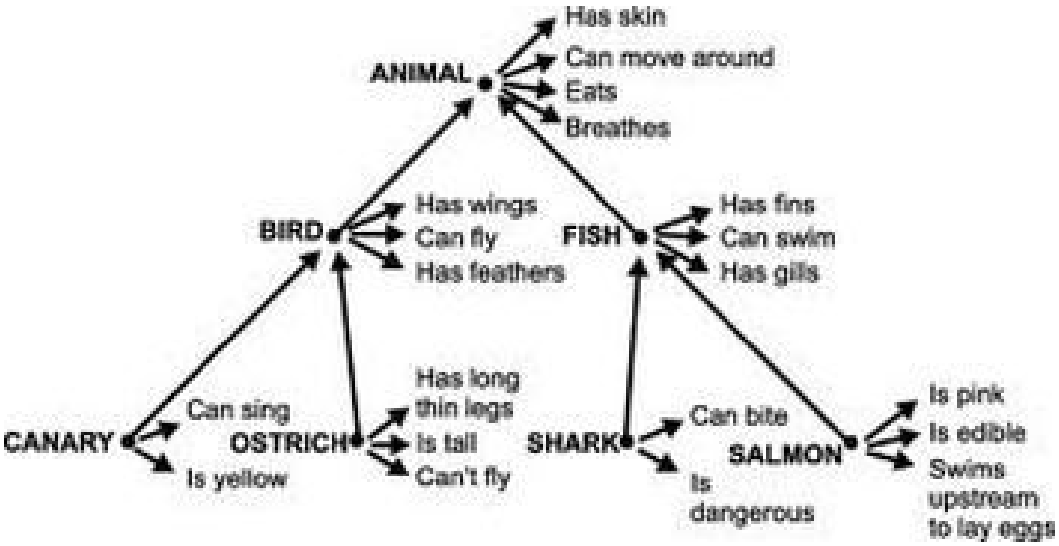

Figure 5.1: Original illustration of the memory semantic network by Collins and Quillian in their *Retrieval time from semantic memory* article [32]

Reusing this human memory representation, Collins and Loftus proposed the spreading activation model to simulate the human memory operations and more particularly the recall of memory units in 1975 [31]. The mental operations are represented in the form of activation levels that spread along the semantic network of the memory. The probability of accessing a memory unit depends on its level of activation. The activation level that passes from one network node to another depends on the semantic relations between them [57]. Such activation level is greater when the memory is accessed frequently or has been accessed recently [151]. In other words the strength of the semantic associations decays over the time [4]. In 1983 Anderson introduced the fan-out effect to enrich the original model: the activation that spreads to one node is distributed among all its neighbors [4]. The latter receive only a fraction of the initial amount of activation. Anderson was also the first, in 1983, to propose the formalization of the level of activation of a node. In [4] he states he gives the following formula:

Preliminary definition 1: (Original spreading activation formula by Anderson). The level of activation of a node y is:

$$a_y = \sum_x f_{xy} a_x + c_y$$

Where:

- c_y is 0 unless y is a focused element in which case c_y is the amount of activation coming from this source.

- f_{xy} determines the level of activation passed the node x to y by taking account of the loss of activation and a *"relative strength"* function depending on the nodes x and y, see [4] for the complete explanation.

Later spreading activation inspired computer sciences algorithms. It was notably used in information retrieval as a processing framework for semantic networks. A lot of variants exist but the core functioning is always the same: first a stimulation value is assigned to one or several node(s) of interest that represent(s) the query and information need. Then this value is propagated to the neighbors' node(s). The values assigned to these neighbors depend on the algorithm purpose, settings and heuristics. During the next iterations the propagation continues from the activated nodes. These iterative propagations are also called *pulses*. This process is repeated till a stop condition is reached e.g. time limit, maximum number of nodes activated or iterations. The graph structure is used as a search controlling pattern and the activation level as a result selection and ranking indicators. With its connectionist nature spreading activation is a typical associative retrieval algorithm. The associative retrieval approaches aim to identify results of interest by relying on their associations with resources that are already known to be relevant.

Spreading activation can be monocentric or polycentric. A monocentric query refers to the initial stimulation of a single origin node. A polycentric query corresponds to the initial stimulation of several nodes at a same time. The formula of spreading activation is presented hereafter.

5.2.2 Core approach

The general spreading activation generic formula, inspired from [1], is presented below:

Preliminary definition 2: (Spreading activation basis)

$$a(i, n+1) = w_s * s(i, n) + \mu f\left(\sum_{j \in Neighbors(i)} w_{ij} * \frac{a(j,n)}{degree_j}\right)$$

Where:

- $a(i, n+1)$ is the activation value of node i at the iteration $n+1$;
- w_s is a weight balancing the stimulation value;
- $s(i, n)$ is the stimulation value of the node i at the iteration n. This value is positive if the node i is an origin node and null otherwise;
- w_{ij} is a weight between nodes i and j;
- $a(j, n)$ is the activation value of the neighbor node j at iteration n;
- $degree_j$ is the degree of the node j. Dividing the activation distributed by the nodes' degree simulates the fan-out effect.
- $Neighbors(i)$ is the set of neighbors of the node i;

- μ is a constant balancing the activation value;
- The function f depends on the algorithm objective. It can be a threshold function for instance.

When it is applied for information retrieval purposes the spreading activation is mentioned in the literature as *constrained* and not *pure*. This term means that a variety of restrictions, decay and stop conditions are used to increase the algorithm relevance and performance. The spreading activation model is fundamentally connectionist. Its application on dense graphs can rapidly lead to the activation of an important amount of nodes. Such *saturation* leads to prohibitive computational costs. It is particularly susceptible to happen in the linked data context where the graphs are very large and connected. Thus it is crucial to apply constraints on the propagation in order to minimize the number of nodes visited and consequently the computational cost. Moreover, it was proved that the *pure* spreading activation technique tends to produce query-independent results [16]. The choice of the constraints is fundamental and deeply influences the outcome of the algorithm. A critical constraint is the weight, w_{ij} in the formula, that determines the amount of activation that passes from one node to another. It has a critical impact on the propagation and by extension on the results. Following weighting functions are frequently mentioned in the literature:

- **Manual weight specification**: in this case manual weights are associated to resources and properties at instance or schema level. This approach hardly scales to the linked data context where the datasets size and heterogeneity prevent efficient manual intervention.
- **Prevalence**: rare resources or properties are considered as more informative and are associated to higher weights, or inversely.
- **Other computations**: a lot of implementations make use of weighing measures obtained from on-purpose processing e.g. semantic similarity, machine learning, usage mining, crowdsourcing or human-based computing approaches can provide a way to compute this value.

Spreading activation is an iterative algorithm that can use a variety of stop conditions. Common ones include:

- **Time limit**: a fixed execution time is specified.
- **Threshold activation value**: a minimum value can be set to trigger the nodes activation. As the propagation spreads it is attenuated and fewer and fewer nodes are activated along the iterations.
- **Number of iterations**: the propagation stops after a determined number of pulses.
- **Number of nodes reached**: the propagation stops when a determined amount of nodes is reached by the propagation.

- **Distance**: the propagation stops when it becomes too distant from the activation source(s), considering the shortest path.
- **Stabilization**: the propagation stops when all the activation values reach a stable state i.e. variations in activations between two iterations are below a given threshold or null.

5.2.3 Information retrieval applications

Early systems Spreading activation was used in the literature to process semantic networks and associative networks i.e. networks having undefined and unlabeled associative relations among the nodes. In [36] the author reviews early works that applied spreading activation for information retrieval purpose. In his Phd thesis Preece (1981) presented the first attempt to use a spreading activation technique for an information retrieval [154]. He implemented it to process a manually-built semantic network constituted of a small set of documents. In [168] (1981), the authors apply a spreading activation algorithm on a semantic network built upon a thesaurus. The system they propose is interactive: it asks the user's agreement on the results at each iteration. An unsupervised version of the algorithm is also available but generates a very long list of results. The GRANT system by Cohen and Kjeldsen (1987) executes spreading activation over a dataset composed of research proposals and potential funding agencies [30]. This semantic network was manually built by several knowledge engineers. The application of a spreading activation algorithm showed positive results especially for "*difficult information needs*" i.e. cases that were difficult to solve using the traditional DBMS. Nevertheless the authors reported misleading results on the "*simple cases*". In his thesis Crestani also mentions several approaches that operate on hypertext data, without mentioning the web. They enable the users "*to search for documents not only by querying a document, but also by browsing it*" e.g. [164].

According to [36] (1997) one of the toughest difficulties for applying spreading activation is to build the semantic network that captures the knowledge covering the desired use-case(s). The semantic networks used by the system during the nineties were often built manually or generated thanks to experimental techniques. They were very expensive to produce. The building automation and maintenance of such semantic networks constituted open research challenges at the time. Of course this problem is not a difficulty in the linked data context we address. Today such semantic networks are available and are already structured. The semantic web graphs renewed the interest about associative retrieval techniques, including spreading activation.

Semantic networks [37] (2000) is one of the earliest works applying spreading activation over a web hypertext graph, thus the links are unlabeled and unweighted. To guide the propagation the authors compute a similarity between the fired nodes and their neighbors. The similarity is based on a cosine compar-

ison between the web pages so-called "*search profiles*". These search profiles are weighted index of terms extracted from the pages content. A users' experiment was performed and showed that the algorithm retrieved relevant results.

In [3] (2003) the authors identify communities of practice in a semantic graph representing the Southampton university' electronic and computer science departments. The dataset contains information about persons, papers, projects and conferences. The system called Ontocopi allows its users to select an instance e.g. a researcher. Then it retrieves a related community of practice thanks to spreading activation. It is possible to use an automatic allocation of weights for the properties. The weights are computed according to the properties prevalence in the dataset. Another possibility is to manually tune it e.g. assigning less activation when the propagation pass the *member of project* or *has author* links. The algorithm spreads in both directions and the propagation stops when it becomes too distant from its origin. The users can tune this distance in order to influence the size of the community of practice retrieved. It is possible to ignore specified instances during the process. This functionality can be used to simulate the effect of losing a research project on the links between the Southampton researchers.

In [114] (2005) the authors describe a method to extend and refine domain ontologies starting from a document corpus. They mine this textual data in order to create a semantic network. To this end they notably use co-occurrence analysis and Wordnet disambiguation. A spreading activation algorithm is used to process this graph in order to find the best candidates to extend the ontology. They use a climate change ontology in their case study. The semantic network has 4 different properties that are manually weighted by the authors. Such weights correspond to an estimated degree of confidence regarding their provenance. It is basically high for the properties belonging to the seed ontology, that have been checked by expert, and less for the automatically extracted ones.

In [111] (2008) the authors perform queries expansion thanks to a spreading activation algorithm in order to build users profiles. A user profile that consists of weighted concepts is built by analyzing the documents that were previously read by the users. This profile is then expanded using spreading activation over a semantic network that contains *is-a* properties. 3 different propagations are executed, a *generalization* one that activates the concept above the stimulated concepts (through the *is-a* properties), a *specialization* one that activate the concept below the stimulated concepts (using again the *is-a* property), and a *relevance one* using the non *is-a* properties. The weights corresponding to these properties are set manually as well as an activation threshold. The maximum number of iterations is set at 2. A users' evaluation showed the benefits of the profile expansion.

In [59] the authors compute similarities between concepts and by extension, documents, using a spreading activation algorithm. They use Wikipedia as a semantic network where the concepts correspond to articles and the hyperlinks to relations between these concepts. The semantic network they propose can be considered as a degraded DBpedia, similar in term of structure but without semantics. The authors also use distance, fan-out and threshold conditions to constrain the

propagation. They proposed and compared 3 weight mappings step and 3 spreading activation strategies to process the graph. They performed experiments to find the best combination. Their method is positively evaluated both for document and concept similarities against 2 others Wikipedia-based measures: the explicit semantic analysis [55] and the Wikipedia link measure [203] approaches.

Semantic web [159] (2004) was the first work that used spreading activation for semantic search. It presents a hybrid approach to search RDF data by combining a keyword-querying system and a spreading activation algorithm. The objective is to use an associative retrieval technique to retrieve results of interest while keeping the intuitiveness of the keyword search queries. The authors first associate a numeric value to each property in the graph using an approach combining a similarity and a specificity measure. These weights are used to modulate the amount of propagation that is passed from one node to another. When the users enter a keyword query it is mapped to a set of nodes in the ontology using the literals attached to them. These nodes are stimulated. The activation paths are also shown to the users for results explanation purposes. Several evaluations on different datasets (university intranet and art-related data) proved the efficiency of the approach. The stop condition is a distance one, set at 3. They also introduce an attenuation factor that decreases the activation values over the iterations.

[79] (2008) proposes a method based on spreading activation to create context-adaptive web applications. The context (e.g. time, weather, devices) of the users as well as the information collection (content) are represented in a single semantic network. When the users' context changes the corresponding node activations values are modified and an activation flows through the network. The content shown on the website is adapted regarding the corresponding activation values. The system takes into account the users' feedback and modifies the weight in consequence. Each user has an individual graph representing his context and content information. This graph serves for personalization and augments the probability to retrieve items of interest over the time. The system uses a maximum number of activated nodes and maximum amount of processed nodes as stop conditions. A users' evaluation confirmed the efficiency of the approach.

In [93] (2008) the authors use spreading activation over a RDF socio-semantic network. The network is a handly-crafted web-crawl composed of 16.468 entities and 25.028 relations. It contains FOAF entities as well as resources that constitute social objects [48] (interests, schools, workplaces, projects, documents). Social objects are objects of interest which indirectly connect several profiles/persons through diverse platform-dependent interactions e.g. share, rate, comment a photo. According to the authors there is a need for new techniques to efficiently navigate the social networks content as they are increasingly large, dense and heterogeneous. Innovative forms of processing are needed within and, even more, across the semantically described social network platforms. In the paper the authors perform monocentric spreading activation to locate sets of nodes that are

closely related to a person using the IBM Galaxy framework[1]. They also execute polycentric queries in order to locate a community centered around several persons. The authors detail the results of such queries but do not evaluate them. The paper is presented as a proof-of-concept showing the application of spreading activation for navigation purposes over socio-semantic networks.

In [90] (2010) the authors apply spreading activation over a personal information management ontology. The ontology covers several types of items such as events, places, activities. When a document is selected the system executes a spreading activation algorithm over the ontology starting from the corresponding instance and its class. It finally suggests a list of related instances and classes that can be used to classify the document. The main interest of the approach is that it implements 3 different formulas influencing the spreading activation. They correspond to short-term, middle-term and long-term memory.

In [53] (2011) the authors answer natural language queries with DBpedia. Their technique combines entity search and a spreading activation algorithm leveraging a semantic relatedness measure. The query is first processed with entity recognition and a parsing technique. It is modeled in the form of a *partial ordered dependency structure*. For instance the query "*from which university did the wife of Barack Obama graduate?*" is transformed in a structured sequence of terms named a partial ordered dependency structure (PODS): e.g. *Barack Obama / wife / graduate / university*. Such terms constitute the starting point of the spreading activation algorithm. The propagation is driven in the graph thanks to a semantic relatedness measure in order to reach nodes that correspond to the terms of the PODS. This semantic measure is the Wikipedia Linked Measure taken from [203]. For instance starting from Barack Obama the propagation will reach the node Michelle Obama because *dbpedia-owl:spouse*, has a high similarity with the next term of the PODS: *wife*. The spreading activation iterates till the size of the PODS is reached. The authors made an evaluation using the QALD dataset challenge[2]. They obtained an average precision of 0.487, an average recall of 0.57 and 70% of answered queries.

The LOD also motivated researches on fast, robust and scalable algorithms for processing RDF data. This is the purpose of the LarKC international project[3] (2011) which developed an open-source and distributed semantic computing platform offering, among others, spreading activation based processing. In [62] the authors detail the performance of a very fast spreading activation application over a large LOD source. They activated millions of nodes in only a few seconds. This impressive result is obtained by executing the algorithm on instances' clusters and not directly on the instances. Nevertheless, the approximation strategies they use are not accurate enough to be used in a knowledge retrieval context like exploratory search as they massively select nodes, do not rank them and do not exploit their semantics finely.

Several approaches based on traversal algorithms mentioned in chapter 4 use

[1]http://www.ibm.com/developerworks/rational/library/into-blue-galaxy/

[2]http://greententacle.techfak.uni-bielefeld.de/ cunger/qald/1/qald-1-challenge.pdf

[3]http://www.larkc.eu/

indeed spreading activation. Lee and al. [109] use it in their associative search system. It is interesting to notice that they propose two result lists by implementing 2 different weighting functions (generality and specificity ones). Spreading activation was also chosen by Fernandez and Heitmann to compute cross-domain recommendations. In the first case [51] the spreading activation algorithm is applied on a small acyclic graph that captures the relation between 2 domains, through chosen properties. In the second case [77] the algorithm is applied on a heterogeneous interest graph that was merged thanks to DBpedia.

5.3 Requirements and motivations

We chose to ground our solution on a spreading activation basis for the reasons presented hereafter. Some of them were obvious before the implementation because they were well-documented, others advantages were observed all along the implementation:

- **It is inspired by human cognition**: there is an interesting resonance between the cognitive psychology origins of the spreading activation and its application in a context of exploratory search. The data can be considered as an external memory and the algorithm retrieves results that should be memorized by the users.
- **It is a well-tried algorithm**: spreading activation has been successfully used in information retrieval since years. Relying on such solid and well-documented method constituted a solid support in the complex contexts of exploratory search and linked data processing.
- **It can takes minimal inputs to produce large result sets**: due to its connectionist functioning spreading activation can be executed starting from minimal and vague inputs e.g. selection of a resource of interest. It is able to retrieve large sets of results, to favor the recall rather than the precision. These aspects are consistent with exploratory search, at least its early stages when the users have a vague idea and are not able to formulate precise queries. They show an orienteering behavior by exploring large sets of potentially relevant result.
- **It is adaptive and flexible**: the weighing function can be easily tuned to cover a variety of use-cases. In this way [109] proposed two weighting functions that produce two result lists for a query. It is possible to allow the users exploring topics from various angles by playing on such weighting function.
- **It is fast**: it has been shown that spreading activation can reach a high level of performance. Some works are especially dedicated to the execution of spreading activation on very large datasets in record times e.g. [62] and [116]. There is also a tradeoff between results quality and execution time. Indeed, spreading activation is a connectionnist and an iterative algorithm so it needs

a vast amount of connections to process during a sufficient number of iterations to produce a relevant result. Otherwise it is possible to design faster but still relevant approximations by relying on fine-tuned, optimized amounts of connections and iterations.

We adapted the original spreading activation formula to propose a semantic adaptation optimized for the processing of large and heterogeneous graphs like the linked data ones. Indeed in this context there is a need to constrain the propagation in order to target relevant parts of the graph only for increasing the algorithm relevance and minimizing its cost of execution. The information need and the corresponding queries are often vague during exploratory search. Thus the algorithm has to identify relevant results starting from simple inputs by extensively relying on the graph richness. The semantic weighting function assumes the task to assign the level of activation that passes from one node to another. In our algorithm it has two effects on the spreading activation process:

- First the activation spreads only to nodes belonging to a subset of classes identified as relevant regarding the topic explored. In other words the *Class Propagation Domain* is a query-dependent semantic filtering operator that identifies the nodes that are eligible to activation ($CPD(o)$, definition 10 below).
- Second it favors the nodes that are similar to the queried node/topic thanks to a triple-based similarity measure ($commontriple(i, o)$, definition 11 below).

To sum up the spreading activation adaptation we propose to retrieve a mix of filtered results that are strongly related and/or similar to the topic explored.

5.4 Monocentric semantic spreading activation function

Spreading activation is monocentric when a single resource is initially stimulated. At the contrary polycentric queries, presented in the following section, have multiple initial stimulations. The queries targeting a unique entity are the most common ones in the context of web search. It has been observed that more than 50% of web search queries are *"pivoting around a single entity that is explicitly named"* [20]. The execution of the monocentric algorithm in the context of the LOD is detail in the chapter 6, section 6.3, page 143.

5.4.1 Formalization

Prior to the algorithm definition, we introduce several necessary definitions on RDF triples as well as the classic graph functions we use:

Definition 1: (RDF triple, RDF graph). Given U a set of URI, L a set of plain and typed Literal and B a set of blank nodes that are *simply indicating the existence*

of a thing, without using an IRI to identify any particular thing[4]. An RDF triple is a 3-tuple $(s,p,o) \in \{U \cup B\} \times U \times \{U \cup B \cup L\}$. s is the node subject of the RDF triple, p the predicate of the triple and o the node object of the triple. An RDF graph is a set of RDF triples.

Definition 2: (RDF typing triple, RDF non-typing triple). An RDF typing triple is a 3-tuple $(s,p,o) \in \{U \cup B\} \times \{rdf:type\} \times \{U \cup B \cup L\}$. An RDF non-typing triple is a 3-tuple $(s,p,o) \in \{U \cup B\} \times \{U \backslash rdf:type\} \times \{U \cup B \cup L\}$.

Definition 3: (Inferred RDF triples, IRDF triples). Inferred RDF triples of an RDF non-typing triple (s,p,o) is the set of RDF triples $(s,p,o) \cup \{(s, \quad rdf:type, t_i), 1 < i < n\} \cup \{(o, rdf:type, c_j), 1 < j < m\}$ obtained after RDFS closure. To ensure that each node has at least one type we give by default the type $rdf:resource$ to each node in accordance with RDF/S semantics.

Let KB be the set of all the triples asserted and inferred in the triple store (def. 1, 2, 3).

Definition 4: (node degree). The node degree of a node j is the number of edges involving the node j:

$$degree_j = |(j,p,y) \in KB \cup (x,p,y) \in KB; x \in \{U \cup B\}; y \in \{U \cup B \cup L\}|$$

Definition 5: (Type depth). The function $depth(t)$ uses the subsumption schema hierarchy (as in RDFS or OWL) to compute the depth of a type t and identify the most precise type(s) available for a node.

$$depth(t) = \begin{cases} = 0 \; if \; t = T \; the \; root \; of \; the \; hierarchy, \\ = 1 + Min_{s_t;(t,rdf:subClassOf,s_t) \in KB} \; depth(s_t) \; otherwise \end{cases}$$

Where the type t is a class in the hierarchy of the RDFS schema and s_t is a direct super class of t in this hierarchy before any transitive closure is computed.

Definition 6: (Node neighborhood) $Neighbor(i)$ is the set of neighbors of the node i:

$$Neighbor(i) = \{x; ((i,p,x) \in KB \vee (x,p,i) \in KB) \wedge p \neq rdf:type \wedge x \in U \cup B\}$$

[4]http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/blank-nodes

Definition 7: (semantic spreading activation: monocentric query).
Here is the formula for a monocentric query i.e. for an interest captured in the form of a unique stimulated resource. A single node is activated:

$$a(i, n+1, o) = s(i, n, o) + w(i, o) * \sum_{j \in Neighbors(i)} \frac{a(j,n,o)}{degree_j}$$

where:

- o is the origin node, the instance of interest initially stimulated e.g. Claude Monet;
- n is the current number of iterations;
- $a(i, n+1, o)$ is the activation value of node i at iteration $n+1$ for an initial stimulation at o;
- j iterates over the neighbors of i;
- $s(i, n, o)$ is the external stimulation value of the node i at iteration n for an initial stimulation at o i.e. 0 if $i \neq o$ and the chosen initial stimulation if $i = o$ and $n = 0$.
- $a(j, n, o)$ is the activation from a neighbor node j of i for a propagation origin o at iteration n;
- $degree_j$ returns the degree of the node j as in the definition 4;
- $w(i, o)$ is a semantic weighting function which takes into account of the semantic properties of the nodes i and o to guide the propagation. $w(i, o)$ will be detailed below.

Definition 8: $Tmax(x)$ is the set of the deepest types t of a given node x according to their $depth(t)$, as in the definition 5:

$$Types(x) = \{t; (x, rdf : type, t) \in KB\}$$

$$Tmax(x) = \{t \in Types(x); \forall t_i \in Types(x); depth(t) \geq depth(t_i); \}$$

Definition 9: $NT(o)$ is a multi-set counting the occurrences of the deepest types in the seed node's neighborhood (as in the definition 6).

$$NT(o) = \{(t, c)\ ;\ t \in Tmax(x)\ ;\ c =\mid \{n \in Neighbor(o)\ ;\ t \in Tmax(n)\} \mid \}$$

Definition 10: The class propagation domain $CPD(o)$ is the set of types through which the propagation spreads. To be precise, the propagation spreads through the nodes which have at least one type present in $CPD(o)$. These classes are selected among the neighbors' classes of the initially stimulated node o. The idea behind $CPD(o)$ is that the most informative types regarding a topic must be

present in its direct neighborhood. A threshold-based filtering operation can be applied to exclude the less prevalent types present in $CPD(o)$. This threshold can also be used to restrain the propagation domain size for performance purpose. This filtering operation aims to apply spreading activation only to an informative graph subset.

$$CPD(o) = \left\{ t; (t,c) \in NT(o);\ \frac{c}{\sum_{(n_i,c_i) \in NT(o)} c_i} \geq threshold \right\}$$

Definition 11: $commontriple(i,o)$ is a similarity measure that aims to improve the algorithm relevance by favoring activation of nodes i having similar properties with the origin o. It is a triple based comparison: the more a node is a subject of triples that share a property p and an object v with triples involving the origin node o as a subject, the more it will receive activation:

$$commontriple(i,o) = \left\{ (i,p,v) \in KB; \exists (o,p,v) \in KB; \right\}$$

Definition 12: $w(i,o)$ is the semantic weighting functionality combining the semantic filtering based on $CPD(o)$ and the similarity enforcement based on $commontriple(i,o)$.

$$w(i,o) = \begin{cases} 0 \; if \; \nexists \; t \; \in \; Types(i);\; t \; \in \; CPD(o) \\ 1 + |commontriple(i,o)| \; otherwise \end{cases}$$

5.4.2 Illustrative example

The illustrative example presented below aims to ease the understanding of the formula. It is freely inspired from DBpedia, the linked dataset that was effectively used for the implementation. Several important steps are illustrated below:

- The first step is to assign the stimulation to the node representing the topic of interest of the exploratory searcher, see Figure 5.2.
- Then the class domain propagation identification starts. The neighbors' deepest types are identified using the *rdfs:subClassOf* properties of the schema, see Figure 5.3.
- In a second time the less prevalent types are excluded from the class propagation domain using a threshold e.g. 0.2, see Figure 5.4.
- Once the class propagation domain is identified the first spreading activation pulse is executed. Only the nodes belonging to the class propagation domain are eligible to activation. The *commontriples* similarity measure, abbreviated *ctples* on the illustration, is computed. In this illustration a fictive property *sim_property* is used for this purpose. The similarity measure is a factor of guidance for the propagation, see Figure 5.5.

- Then the second pulse is executed, all the nodes having a positive activation value propagate this value to their neighbors. The illustration on Figure 5.6 shows notably the facts that the similarity measure is always computed regarding the origin of activation. In a lot of spreading activation implementation the semantic weighting function consider the nodes that spread and receive the activation, and not the origin.
- Then several pulses of spreading activation are executed till a stop condition is reached e.g. a maximum number of iterations. The top-k most activated nodes constitute the results. We can use the class propagation domain to sort the results, see Figure 5.7.

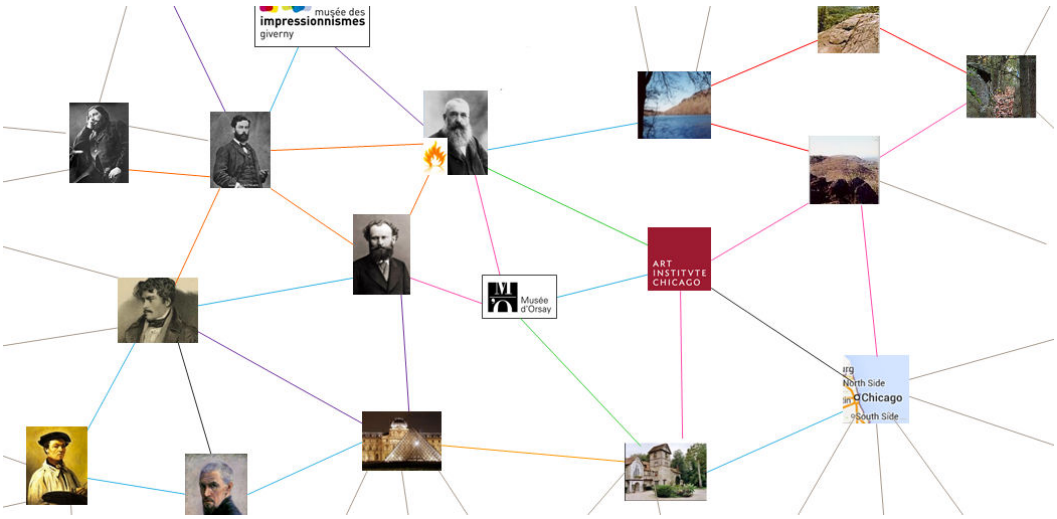

Figure 5.2: The node Claude Monet is stimulated

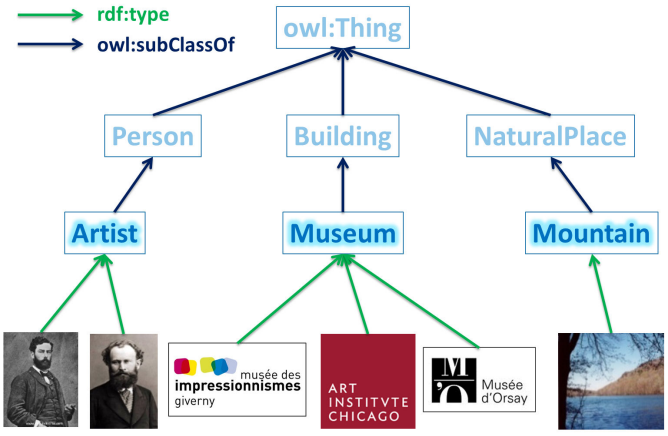

Figure 5.3: During the class propagation domain computation the first step is to identify the neighbors' deepest types

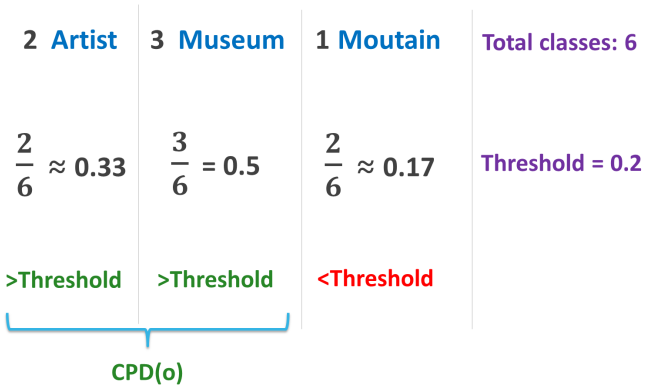

Figure 5.4: A threshold function is used to exclude the less prevalent types from the class propagation domain

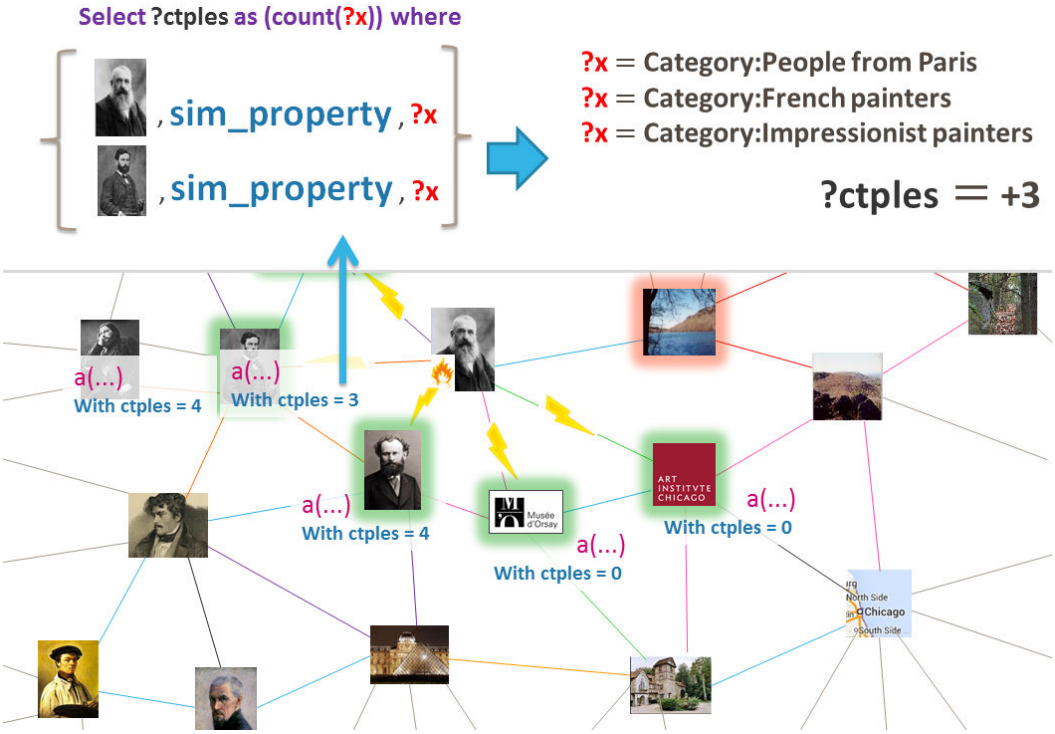

Figure 5.5: Commontriples similarity measure (*ctples* here) computation

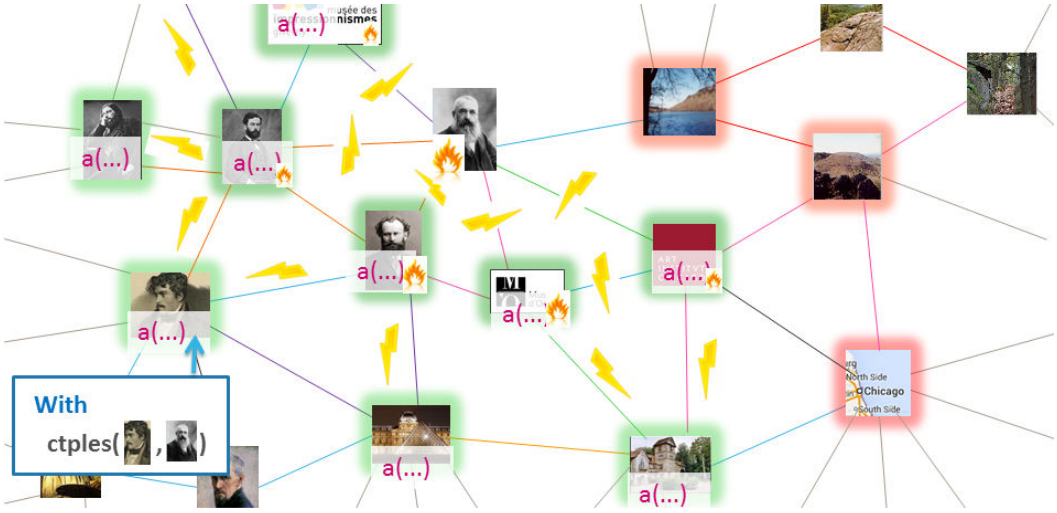

Figure 5.6: Second iteration of spreading activation

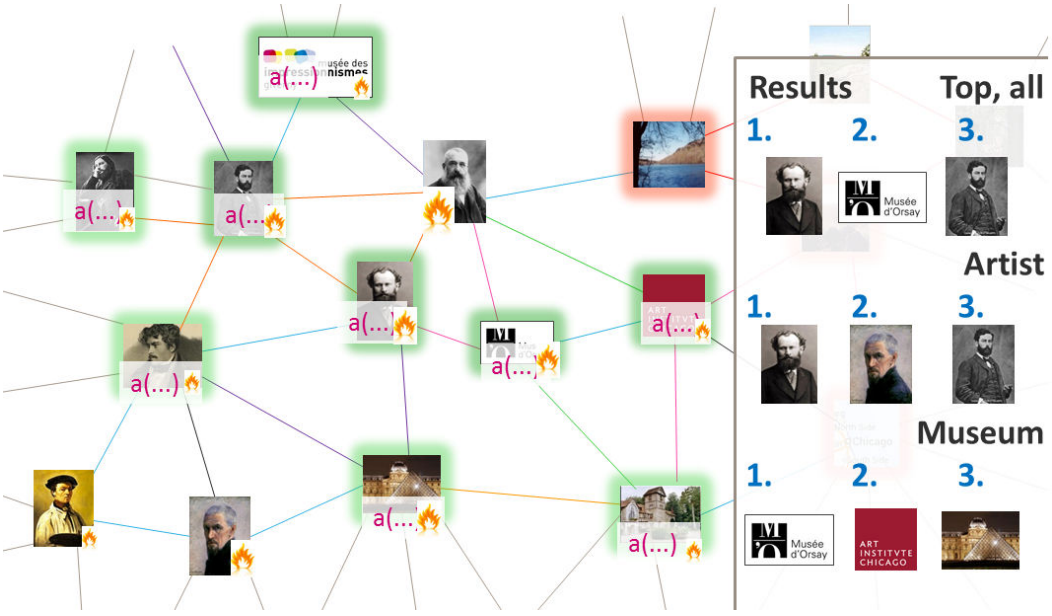

Figure 5.7: When a stop condition is reached the top-k most activated nodes constitute the result set

5.5 Polycentric semantic spreading activation function

The linked nature of the LOD graphs offers a powerful support to enable explorations having the following form *knowing the user interest for X, Y and Z what can he discover/learn which is related to all these resources?* We call these queries *composite interest queries*: they combine several unitary interests in order to explore a connected result space that is at their crossroad. Even more the linked datasets offer a valuable ground to solve such queries starting from heterogeneous inputs, having diverse types and/or belonging to various domains. For instance a user might be interested by making discoveries related to both The Beatles and the movie director Ken Loach[5]. At an algorithmic level such information needs are solved by polycentric spreading activation where the unitary resources of interest constitute the set of initial stimulations. The execution of the monocentric algorithm in the context of the LOD is detail in the chapter 6, section 6.4, page 152

5.5.1 Formalization

The result of a polycentric query is the product-intersection of several monocentric propagations results (definition 7). Consequently, to be activated at a polycentric level a node has to be reached by the propagations starting from all the origins. The level of activation of a node is divided by the logarithmic function of its degree, this point is explained below:

Definition 13: (Semantic Spreading Activation algorithm, polycentric query)

$$a_{poly}(i,n) = \prod_{o \in O}[a(i,n,o)]/log(degree_i)$$

where:

- O is the set of nodes initially stimulated;
- $a_{poly}(i,n)$ is the polycentric value of the node i: the product of the activation values of i regarding the various propagations spreading at the iteration n, differentiated by their origin o. The product is used instead of the sum in order to avoid a potential disequilibrium provoked by the differences in the monocentric activations distributions. Indeed, the initial amount of activation passed to the neighbors is divided by the origin degree. Thus the amount of activation per node can be very different regarding the degree of the propagation origin. The division by $log(degree_i)$ aims to minimize the importance of the highly connected nodes that can be present in the monocentric propagations intersections but not very informative;

[5]This example was the first query entered by a user in the Discovery Hub application

- $a(i,n,o)$ is the activation value of the node i at iteration n for a spreading activation taking its origin at o, as in the definition 7.

Definition 13: $NT(O)$ is the union of the $NT(o)$ multisets with $o \in O$ used for polycentric queries.

$$NT(O) = \bigcup_{o \in O} NT(o)$$

Definition 14: $CPD(O)$ is the classes propagation domain, taking into account the semantics of all the initially stimulated nodes $o \in O$:

$$CPD(O) = \left\{t; (t,c) \in NT(O);\ \frac{c}{\sum_{(n_i,c_i) \in NT(O)} c_i} \geq threshold\right\}$$

During polycentric queries $w(i,o)$ is modified as follows in order to favor the identification of polycentric results using $CPD(O)$:

$$w(i,o) = \begin{cases} 0\ if\ \nexists\ t\ \in Types(i);\ t \in CPD(O) \\ 1 + |commontriple(i,o)|\ otherwise \end{cases}$$

5.5.2 Illustrative example

The graph on Figure 5.8 illustrates the fact that some LOD datasets are cross-domain and very heterogeneous. They connect a high variety of resources in a very dense structure and can support a wide range of composite queries. For instance a user might be interested in exploring the artistic scene and cross-influences existing between the film director *Ken Loach* and the band *The Beatles*. He might also be interested in the relations between *Ken Loach* and *Margaret Thatcher*, as the Ken Loach movies are politically engaged and depicts social and political contexts through personal stories; he is *"known for his naturalistic, social realist directing style and for his socialism, which are evident in his film treatment of social issues*"[6]. We will use these two examples of composite queries to illustrate the algorithm behavior in case of polycentric queries.

- The figure 5.9 illustrate the shared class propagation domain computation for polycentric queries. When the resource Ken Loach is combined with The Beatles the artistic facet they share is captured by the $CPD(O)$. When it is combined with Margaret Thatcher the politics aspects take precedence over the artistic ones. Note that the real queries $CPD(O)$ for an implementation on DBpedia are presented in the *discussion* section of this chapter.
- It is observable on Figure 5.10 and 5.11 that the shared class propagation domain guides the propagation in part of the graphs that are relevant regarding the unitary interest. On Figure 5.10 the node *Looking for Eric* is included in the class propagation domain but is distant from Margaret Thatcher. It will

[6]http://en.wikipedia.org/wiki/Ken_Loach

not receive a sufficient amount of activation to be part of the top results. This specific case illustrates the interest of using a product rather than a sum: the nodes that are close to only one origin have a low or null amount of activation at polycentric level.

- Then the spreading activation pulses are performed. To be precise, two independent propagations spread over the graph, distinguished by their origins. The nodes that are activated by the two propagations constitute the polycentric results, see Figure 5.11. On this illustrative example (Ken Loach and The Beatles) the *Who* band node polycentric value is null because it is only activated by the propagation originating from The Beatles. It is also observable that the *Looking for Eric* node is better ranked than the *Rock* one. It is due to the division by $log(degree_i)$. The node *Rock* represents a very popular music genre, it is well connected in the graph. It is more probable that a node is connected to *Rock* than to *Looking for Eric*. In other words, *Rock* is less informative according to the information theory, the division by $log(degree_i)$ aims to lower its rank.

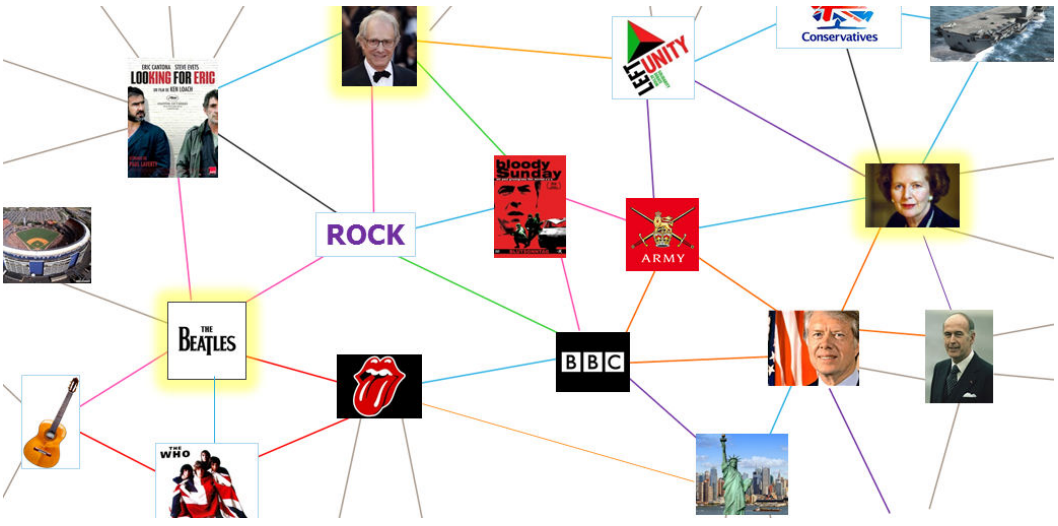

Figure 5.8: Some linked datasets are very heterogeneous and constitute valuable supports for cross-domain information need solving

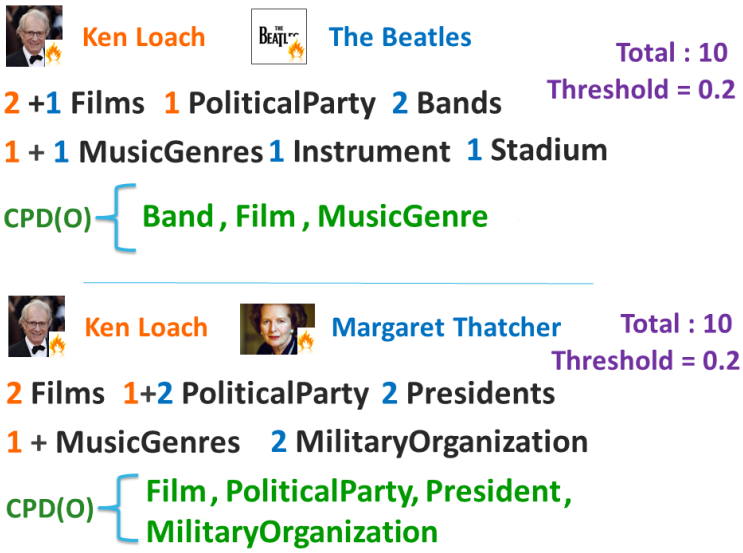

Figure 5.9: Shared class propagation domain computation in case of polycentric queries

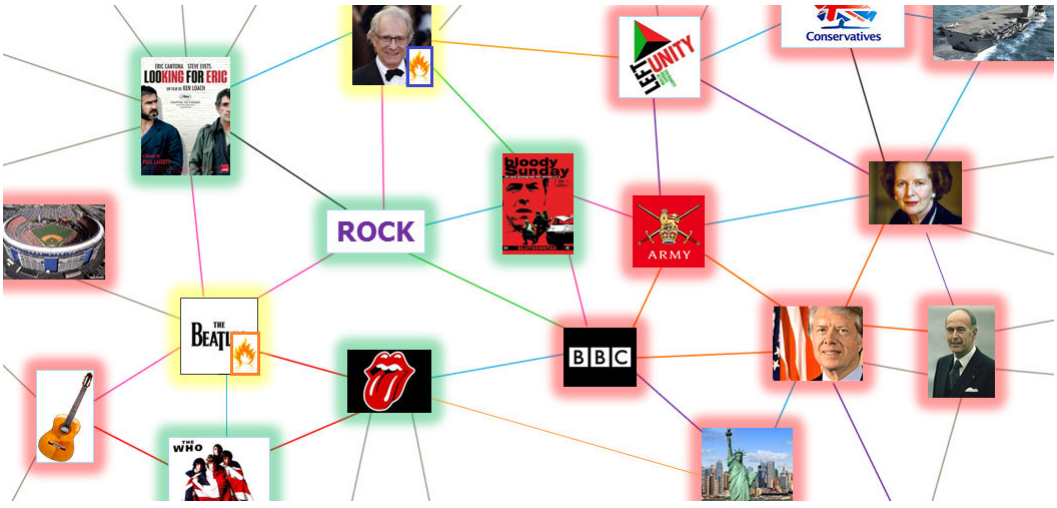

Figure 5.10: Class propagation effect for the query combining Ken Loach and The Beatles: the classes inside the CPD are circled in green

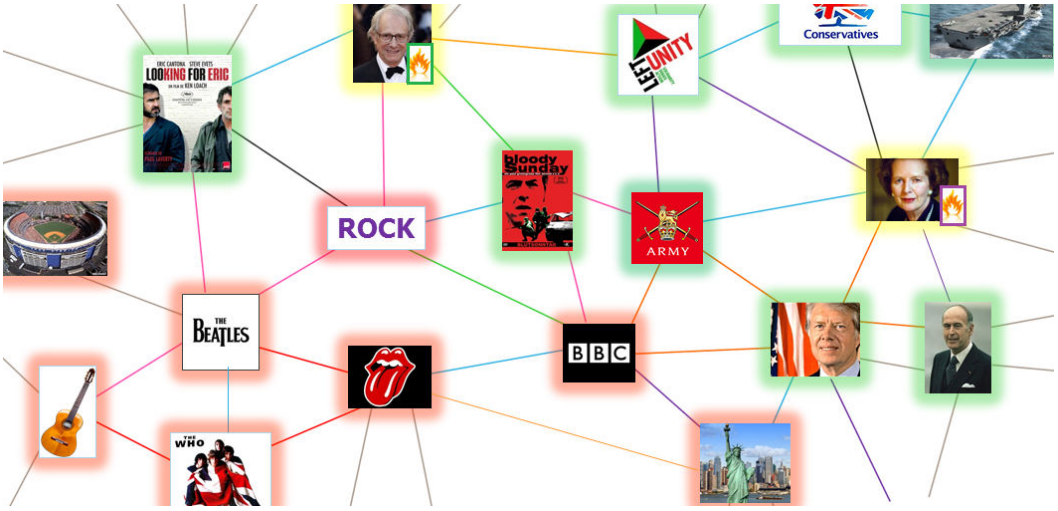

Figure 5.11: Class propagation effect for the query combining Ken Loach and Margaret Thatcher: the classes inside the CPD are circled in green

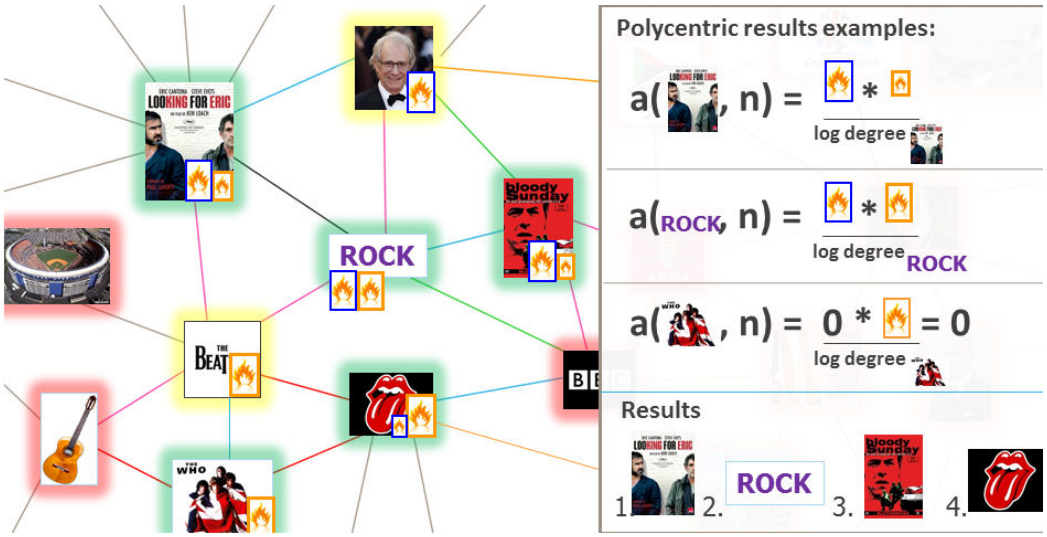

Figure 5.12: Illustration of some polycentric results computations

5.6 Advanced querying functions

A limit of existing algorithm-based linked-data exploration systems identified in chapter 4 is they often offer only one exploration perspective i.e. the users can not or hardly influence the query results in a *direction* of interest. In this thesis we support the idea that a plurality of relevant exploration perspectives on a topic can be offered to the users. Indeed, the objects described in linked datasets can be rich, complex and approached in many manners. For example, the users can be interested in a painter in many ways: works, epoch, movement, entourage, social or political contexts and more. The users may also be interested by basic information or by unexpected ones depending on their actual knowledge about the painter. To sum up a single interest can be explored through many perspectives corresponding to different *knowledge nuances*. In the graph context of linked data these perspectives correspond to different non-exclusive sets of objects and relations that are informative on a topic regarding specific facets of interest. The two algorithm variants presented below aim to unveil hardly identifiable knowledge nuances and to expose information that can be hidden by the basic algorithm:

- The criteria of interest specification variant (definition 15 below).
- The controlled randomness injection variant (definition 16 below).

5.6.1 Formalization

Criteria of interest specification The criteria of interest specification variant aims to unveil results that are related to topic facets of interest. For instance a user might be interested discovering the French non-impressionist painters related to Claude Monet. He might want later to discover the non-French painters that were impressionist and related to Monet. To solve these information needs the similarity function is modified to only process targeted triples' patterns (set of properties and objects), and to allow to associate them a positive or negative weight. To reuse the previous example the triples that capture the fact that a painter is impressionist can receive a negative weight whereas the triples specifying that a painter is French receive a positive weight. In this case the similarity with French and/or non-impressionist painters will be higher. In other words such painters will be more activated. This algorithm variant aims to drive the propagation and some specific parts of the graph.

Definition 15: (semantic spreading activation, criteria of interest specification variant)

$$a_{criteria}(i, n+1, o, C) = s(i, n, o) + w_{crits}(i, o, c) * \sum_j \frac{a(j,n,o)}{degree_j}$$

where:

- C is a multiset containing the criteria of interest c and their assigned values v e.g. *French* with value 1 and *Impressionist* with value -1:

$$C = \{(c, v)\}$$

with:

$$w_{criteria}(i, o, c) = \begin{cases} 0 \; if \;\; t \in Types(i) \; ; \; t \in CPD(o) \\ 1 \; + \; \sum_{c \in C} commontriple_{criteria}(i, o, C) \; | otherwise \end{cases}$$

with:

- $commontriple_{criteria}(i, o, C) = v * commontriple(i, o, c)$ with $(c, v) \in C$.
- $commontriple(i, o, c) = \{(i, p, c) \in KB; \exists (o, p, c) \in KB; p \in P\}$.
- P is the set of properties used to compute the similarity, chosen by the developer.

Controlled randomness It is possible to inject randomness into the activation value computation in order to modify the ranking scheme and expose unexpected results. We voluntarily change the algorithm behavior that is originally designed to retrieve the results that are the most relevant possible, in other words the most obvious. This operation is particularly interesting for the experts that want to retrieve unusual information in order to deepen their peripheral knowledge on a topic. Moreover: *"for exploratory searches, there is novel value in encountering pages not frequently visited by other users of the system. These pages may contain information that yields unique insights or competitive advantage"* [194].

To avoid to confront quickly the user with too surprising results the randomized version of the algorithm is different if the chosen level of randomness is inferior and equal or superior to 0.5 (with a minimum randomness value at 0, and a maximum at 1). If the value is inferior or equal to 0.5 the results are randomized only at the last iteration. In other word the spreading activation occurs normally till the last iteration. If the desired randomness level is superior to 0.5 the randomization occurs at each iteration influencing strongly the spreading activation algorithm and consequently the results list. An interest of the randomized variant is that it is divergent and non-deterministic, it produces different results every times:

Definition 16: (semantic spreading activation, controlled randomness variant)

$$a_{random}(i,n,o,r) =$$

$$\begin{cases} (1-r) * a(i,n,o) + r * random() \; if \; r > 0.5 \\ a(i,n,o) \; if \; r \leq 0.5 \; and \; n < maxPulse \\ (1-r) * a(i,n,o) + r * random() \; otherwise \end{cases}$$

where:

- r is the level of randomness desired, comprised between 0 and 1;
- $random()$ retrieves a random value between 0 and 1;
- $maxPulse$ is the maximum number of spreading activation iterations, the choice of this stop condition is discussed in the sub-subsection 6.3.2.2 on page 147.

5.6.2 Illustrative examples

The figure 5.13 illustrates the criteria of interest specification variant. The association of positive, null and negative weights on the Claude Monet properties-values (e.g. *Artist from Paris*, *French painters*) modifies the level of similarity of the activated nodes. Consequently different parts of the graph are explored regarding the assigned values. The ranking and the composition of the results are impacted.

The figure 5.14 illustrates the behavior of the randomized variant with a level of randomness comprised between 0 and 0.5. In this case the randomization is applied as a post-processing operation. Each resource activation value is multiplied by a randomized value between 0 and 1, in other words the basis results list is re-ranked. Resources can enter and get out of the top results list.

When the level of randomness is superior to 0.5 the activation values are randomized at the proportion of the threshold at each iteration, see Figure 5.15. It results in a strong modification of the results composition and ranking as the propagation follows unusual paths.

5.7 Discussion

It is important to stress that the absence of solid exploratory search golden truth prevents the comparison of a vast amount of algorithm configurations. As the human perception is critical during exploratory search we validated our formulas thanks to in-depth user evaluations (see chapter 8). These algorithms were shaped after several rounds of empirical observations and according to the findings in the existing literature. The most critical algorithmic design choices are explained hereafter. The initial idea was to use the semantic similarity function offered by

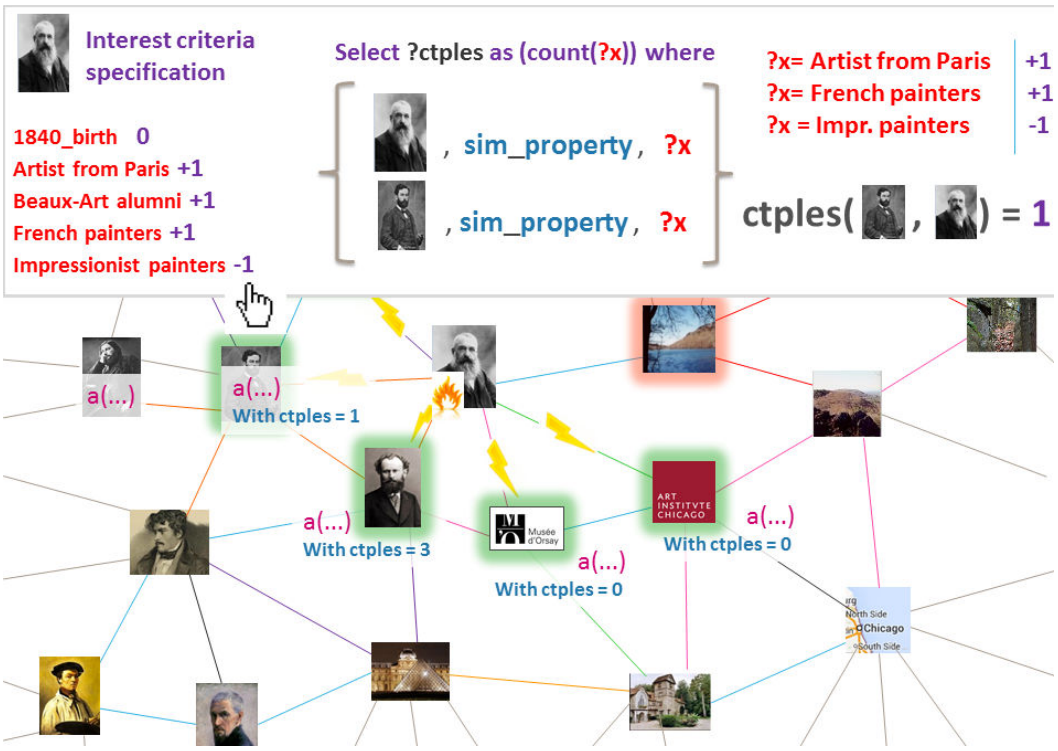

Figure 5.13: The specification of criteria of interest influences the algorithm

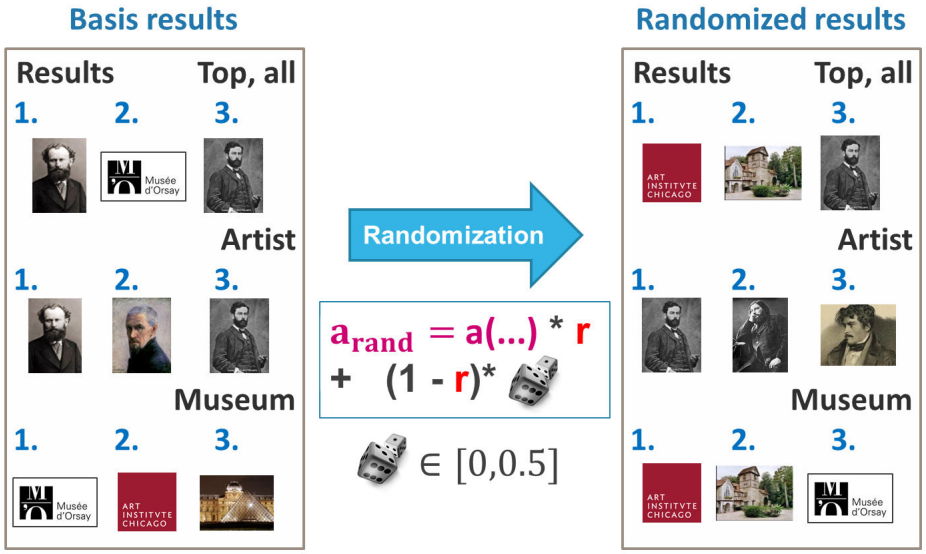

Figure 5.14: For a level of randomness inferior or equal to 0.5 the activation values are only randomized once, after the basic spreading activation process

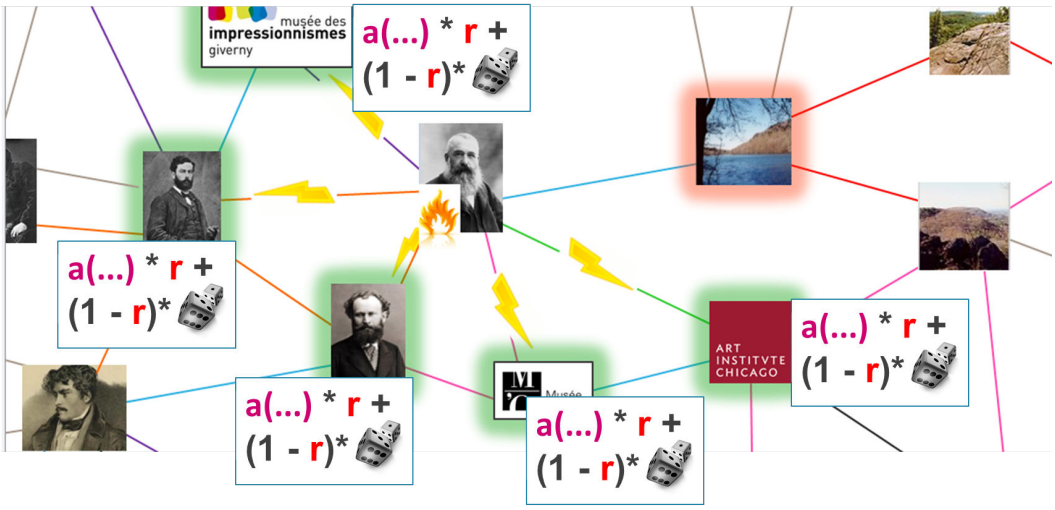

Figure 5.15: Illustration of the algorithm behavior for a randomness value superior to 0.5, with r being the desired level of randomness

the Kgram[7] inference engine for the semantic weighting function. This function is formally described in [33]. It computes the semantic similarity between two classes using the ontological schema subsumption links. The idea behind this measure is explained below:

> *Starting from the fact that in an ontology, low level classes are semantically closer than top level classes (for instance TechnicalReport and ResearchReport which are brothers at depth 10 are closer than Event and Entity which are brothers at depth 1), we want the ontological distance between types to decrease with depth: the deeper the closer.* [34]

Several problems arose when this similarity measure was implemented as the spreading activation weighting function. They are listed below:

- The schema of the linked data sources actually available are generally simple and have a low depth. It limits the efficiency of such purely schema-based similarity measure.
- This similarity measure can only be applied on instances that have at least one class specified. This is not always the case in the LOD context where many datasets have untyped instances, e.g. in DBpedia.
- This semantic measure operates at the schema level and is not instance dependent. Some instances can share the same type but be actually very different. This point is critical and constituted the main motivation to develop the class-propagation domain filtering functionality. It is extensively discussed below.

[7] https://wimmics.inria.fr/node/26

The results obtained with the Kgram semantic similarity measure led to the following important observations. Schema-level knowledge should be used to guide the propagation and structure the results space. To do so an identification of informative types regarding the explored topic have to be performed. A pure class-similarity metric is not satisfying because it masks numerous aspects. A relatedness measure is more adapted i.e. it is obvious that paintings and museums are relevant regarding painters nevertheless the relation between them can not be captured by a semantic similarity. Painters are related to museums, not similar to them. Later during the thesis the use of the encyclopedic knowledge patterns (EKP), implemented by Aemoo, was considered. The EKPs capture such relatedness between classes. Nevertheless, like the Kgram similarity measure, it is class-dependent. It means that first it can not be used for untyped resources. Second all the instances of a class share the same EKP when they can be significantly different.

Thus we designed a specific semantic weighting function where the identification of informative types is performed at instances level and specific to the query ($CPD(o)$). It is based on the idea that informative types about a resource should be present in its direct neighborhood. Comparisons between the Kgram semantic similarity function, the EKP and the DBpedia class-propagation domain for the Claude Monet and Eugène Delacroix resources are shown in the table 5.1. It is observable in this table that the two CPDs overlap regarding artistic aspects (*Artist, Museum* classes) but also contain types that are specific to each painters. Such nuances are impossible to identify with the Kgram similarity function nor the *Artist* EKP:

- The types *River* and *Plant* are specific to Claude Monet who was particularly inspired by the nature, notably in the famous Nymphéas (water lilies) series[8]. The type *Disease* can appear morbid in the first place but captures a very interesting facet of Monet. The painter was suffering from the cataract disease. The progression of the disease had an important impact over its artworks: *"the paintings done while the cataracts affected his vision have a general reddish tone, which is characteristic of the vision of cataract victims"*[9].
- The types *MilitaryConflict* and *PrimeMinister* are specific to Eugène Delacroix who was politically engaged in favor of Napoleon and who painted important historical events. These works include *"la Liberté guidant le peuple"* (*Liberty leading the people*) which became an iconic representation of the French Revolution. This artwork was used by the band Coldplay[10] for the cover of their album *Viva la vida or Death and All His Friends*. It also inspired one French singer video clip[11]. Others music albums used Delacroix artworks

[8] http://www.musee-orangerie.fr/homes/home_id24799_u1l2.htm
[9] http://en.wikipedia.org/wiki/Claude_Monet#Failing_sight
[10] http://www.coldplay.com/
[11] http://en.wikipedia.org/wiki/D%C3%A9senchant%C3%A9e

for their covers[12] [13]. The fact that the painter was an inspiration for musical artists explains the presence of the *Single* and *Album* types. The presence of the type *Planet* is curious and related to the 10310 Delacroix asteroid[14], named after the painter.

Claude Monet	**DBpedia Artist type**		**Eugène Delacroix**
AdministrativeRegion	**Kgram similarity**	**EKP**	**AdministrativeRegion**
Artist	Actor (0.9248)	Actor	Album
Book	Comedian (0.9248)	Administrative Region	Artist
Building	ComicsCreator (0.9248)	Adult Actor	Book
Company	MusicalArtist (0.9248)	Album	BritishRoyalty
Disease	Writer (0.9248)	Artist	Building
Film	AdultActor (0.8913)	Award	Film
Museum	VoiceActor (0.8913)	Band	HistoricPlace
Person	Person (0.8602)	Beverage	MilitaryConflict
Plant	Ambassador (0.7547)	Book	Museum
River	Architect (0.7547)	British Royalty	Person
TelevisionShow	Astronaut (0.7547)	Broadcast	Planet
Town	Athlete (0.7547)	Building	PrimeMinister
University	BritishRoyalty (0.7547)	City	School
Writer	Celebrity (0.7547)	College	Single
	ChessPlayer (0.7547)	Comedian	Town
	etc.	etc. (A total of 68 classes)	Writer

Table 5.1: Class propagation domain, Kgram similarity measure and encyclopedic knowledge pattern of the Claude Monet and Eugène Delacroix resources

Another example that particularly shows the interest of an instance-based informative classes identification is given in the table 5.2. In DBpedia both the civil and military aircrafts share the type *Aircraft*. The table hereafter presents the DBpedia class propagation domains of the Boeing 747 and B17. They are respectively the first civil and military Aircraft resources by decreasing degree order in DBpedia. It is observable in this table that the civilian (*Airline*) and military (*MilitaryConflict*, *Person*, *Unit*, *Ship*) contexts of each plane are well-identified. *MilitaryUnit* is also present in the 747 CPD as a military version of the plane was proposed.

As mentioned before the fact the class propagation domain is instance-based also enables the identification of a shared context in the case of composite queries. To illustrate this point we reuse the composite query examples implying Ken Loach. It is observable below that the shared class propagation domains identify music-related types for a query combining Ken Loach and The Beatles (*MusicGenre*, *RadioStation*, *Song*). It identifies types related to politics for a query combining Ken Loach and Margaret Thatcher (*Election*, *MemberOfParliament*, *MilitaryConflict*, *OfficeHolder*, *PoliticalParty*, *Politician*, *President*).

Later we re-introduced a similarity measure, with the idea that results that are similar to the topic explored are interesting to the exploratory searchers. Exposing similar results might notably reinforce their confidence in the system. The measure

[12] en.wikipedia.org/wiki/Prospekt's_March
[13] http://en.wikipedia.org/wiki/The_IVth_Crusade
[14] http://en.wikipedia.org/wiki/10310_Delacroix

Boeing 747 (civil)	DBpedia Aircraft type		Boeing B-17 (military)
Class propagation domain	**Kgram similarity**	**EKP**	**Class propagation domain**
Aircraft	MeanOfTransportation (0.8602)	Aircraft	Aircraft
Airline	Automobile (0.7547)	Airport	Airport
Airport	Instrument (0.7547)	City	Company
Company	Locomotive (0.7547)	Country	Film
Film	Rocket (0.7547)	MilitaryConflict	MilitaryConflict
MilitaryUnit	Ship (0.7547)	MilitaryUnit	MilitaryPerson
Museum	SpaceShuttle (0.7547)	Weapon	MilitaryUnit
Person	SpaceStation (0.7547)		Museum
Single	Spacecraft (0.7547)		Person
VideoGame	Weapon (0.7547)		Ship
	AcademicJournal (0.606)		VideoGame
	Activity (0.606)		Weapon
	Activity (0.606) etc.		

Table 5.2: Class propagation domains of the Boeing 747 and Boeing B-17 resources

First resource	**Ken Loach**		
Second resource	**The Beatles**	**Margaret Thatcher**	
Class propagation domain	Album	Album	PoliticalParty
	Band	Book	Politician
	Film	Company	President
	MusicalArtist	Election	School
	MusicGenre	Film	Single
	Person	MemberOfParliament	TelevisionShow
	RadioStation	MilitaryConflict	University
	Single	MusicalArtist	Writer
	Song	OfficeHolder	
	TelevisionShow	Organisation	
		Person	

Table 5.3: Class propagation domain for 2 composite queries implying Ken Loach, one with The Beatles and the other with Margaret Thatcher

we use is very simple. It is agnostic to the data and to the informational domain. It is also very cheap to compute. It is in line with the important concerns about the implementation computational costs. This last point will be developed in the next chapter.

5.8 Conclusion

In this chapter we presented the spreading activation algorithm, it origins, basic formula and its applications for information retrieval including various forms of semantic search. We detailed the formalizations for monocentric, polycentric queries as well as the criteria of interest and randomized variants. The objective of proposing several algorithms is to offer a battery of possible queries allowing to deeply explore a topic, from several angles. We illustrated such formalizations with examples inspired from DBpedia in order to ease their understanding. Finally we discussed the main algorithm design choices we made. In the following chapter we present the implementation we built on top of DBpedia.

CHAPTER 6

Remote semantic spreading activation by incrementally importing distant triples

Contents

6.1 Introduction

In this chapter we characterize and provide the design rationale of all the algorithms previously formalized: the monocentric, polycentric queries as well as the criteria of interest specification and controlled randomness injection variants. Such

implementations solve the problem of allowing multi-perspective explorations on topics. It notably unveils hardly identifiable knowledge nuances using rich but complex data sources. This chapter particularly addresses the question of how to flexibly process the LOD[1]. It is more specifically focused on how to leverage its distributed aspect and how to control the algorithm response-time while preserving the quality of its results.

In order to overtake the state-of-the-art algorithm-based approaches and their lack of flexibility we made two important design choices. First the results are not pre-stored; they are computed at query-time. Second the data used for the computation are distant and consumed remotely from online SPARQL endpoints. This software architecture meets our requirement of flexibility in terms of data exploration. Indeed if the results are computed at query-time it allows the user to influence important algorithms parameters e.g. specifying his criteria of interest and/or declare a desired level of randomness/unexpectedness. Moreover the LOD is distributed by nature. As more and more linked data sources are available on the web it is interesting to address them remotely for (exploratory) search.

DBpedia, or more precisely the DBpedia chapters, were chosen for the implementation. DBpedia is a cross-domain knowledge source. Its coverage in terms of topics makes it ideal to build a generic exploratory search engine. It can support a wide range of information needs, including ones which necessitate a cross-domain computation e.g. heterogeneous composite queries. We also wanted to leverage the DBpedia internationalization process outcomes. Indeed DBpedia local chapters exist now in 15 languages[2] e.g. the French[3], German[4], Italian[5] and Spanish[6] ones. They follow the LOD principles and constitute together a distributed knowledge base. Each one has its own web domain, data graph and SPARQL endpoint. Equivalences between them are specified using the *owl:sameAs* property. We consequently target a subset of the LOD instead of a single dataset. A specific interest of the DBpedia chapters is that the knowledge they contain overlap to a certain extent. For instance *Claude Monet* is described in all the DBpedia chapters. But the *local* descriptions often reflect cultural elements that are related to a language or a country. The DBpedia chapters act as cultural prisms through which the topics are described.

In order to reach our double objective of computing the results at query-time from distant data we couple the algorithm to an incremental import technique. Along the spreading activation iterations the code replicates a sub-graph from a targeted SPARQL endpoint in a local triple-store. This approach is extensively discussed in this chapter for both the monocentric and the polycentric queries. The

[1]The author especially thanks Olivier Corby for his precious help on the Kgram inference engine and SPARQL querying

[2]http://dbpedia.org/Internationalization

[3]http://fr.dbpedia.org

[4]http://de.dbpedia.org

[5]http://it.dbpedia.org

[6]http://es.dbpedia.org

main idea is that the neighbors of the most activated nodes are imported at each iteration, until a limit is reached. The spreading activation can reach more nodes of interest by following relevant paths of propagation, but its expansion over the graph is highly controlled.

This approach offers the flexibility we want for the SPARQL endpoint selection and for the choice of the computation parameters. Nevertheless it brings also major research questions regarding the triangular trade-off it introduces between the size of the sub-graph imported, the algorithm cost of execution and the quality of the results. In this chapter we focus on the relation between the size of the import and the cost of execution which are crucial to demonstrate the feasibility of the approach, independently of the quality of the results. We present the extensive analysis we performed to understand the algorithm implementation behavior on DBpedia. The choice of its important computation parameters, particularly the import size and the number of iterations, are discussed. The quality of the results is out of the scope of this chapter and will be addressed by users' experimentations in chapter 8.

In this chapter we will review (6a) the implementation requirements, the chosen software architecture and settings as well as the datasets used, (6b) the monocentric queries implementation and their calibration, (6c) the polycentric queries implementation and their calibration, (6d) the advanced queries implementation, (6e) the analysis performed to discuss the applicability of the algorithms outside of the DBpedia context. The implementation and corresponding analyses of the monocentric, polycentric and advanced querying variants were respectively published in [128], [123] and [126].

6.2 Algorithmic and architectural design

This subsection details first the implementation requirements. Second, it describes the software architecture and focuses more precisely on the coupling between the spreading activation execution and the sub-graph importation. Third several settings that did not require an analysis are presented. Fourth the choice of the DBpedia dataset(s) as the knowledge source is motivated and some of its important characteristics are presented.

6.2.1 Requirements

The introduction mentioned the 2 main requirements for the framework implementation: its capability to process distant data as well as to compute the results at query-time. The main motivations behind these requirements are detailed below:

The framework has to process remote linked data. Being able to leverage the distributed aspect of the LOD offers several decisive advantages regarding the query processing. It is also consistent with the changing nature of the LOD and its *potentially infinite* size:

- It offers the possibility to target the most descriptive linked dataset regarding a query and even merge them. In the context of DBpedia, the local chapter offering the most precise description for a topic can be used. It can be done by comparing the resources' degrees in the different chapters for instance.
- Linked datasets can be voluntarily selected to produce a specific result. As mentioned before the DBpedia local chapters can be used as cultural prisms. Exploring Monet in the German DBpedia chapter is more likely to unveil its connections with German artists and museums. Moreover some LOD datasets are highly specialized and can be used for solving precise information needs e.g. *Drugbank*[7] for pharmaceutical exploratory search.
- It allows to target several data sources at a time in order to solve complex information needs. For instance it can be used to offer a more complete view on a topic, or to fill a knowledge gap in a dataset with triples coming from another one.
- It allows to use data from commercial providers that offers API accesses rather than complete dumps[8]. The increasing success of commercial data providers in the close future seems more and more probable according to recent researches [207]. Yahoo revealed they partially built their Knowledge Graph using data from payed providers [20].

The framework has to compute the results at query-time: being able to compute the results on-demand allows to reach a new level of flexibility in terms of data processing. It opens the door to new exploratory search use-cases:

- As we do not target a specific domain it is difficult to compute and store all the queries' results such as the polycentric combinations or all the possible criteria of interest specification choices. If we consider the fact that several SPARQL endpoints can be queried such pre-computation appears impossible. It has a CPU and storage costs that makes it unfeasible, an approach computing the results on-demand appears more pragmatic.
- The linked datasets compositions are constantly changing due to updates. One illustration of such dynamics is the live version of DBpedia[9]. This initiative captures in RDF the changes that continuously occur in Wikipedia. It provides triples change-sets to keep a DBpedia implementation up-to-date [10]. Indeed the main DBpedia extraction is performed periodically and its content and coverage of certain topics can be rapidly outdated. Several use-cases of exploratory search necessitate data freshness, such as journalism for instance.
- The knowledge graphs are not the only possible context of application for the proposed algorithms. Social networks, where heterogeneous data change in

[7] http://datahub.io/dataset/fu-berlin-drugbank
[8] e.g. https://developer.seevl.fm/pricing
[9] http://live.dbpedia.org/
[10] http://live.dbpedia.org/changesets/

real-time, is another realistic use-case that justifies the need for query-time processing.

To meet our requirements we need to elaborate a lightweight and flexible data processing implementation. Such approach will be in line with the semantic web vision where the data distribution is a key aspect. However it raises significant challenges in terms of control of the computational cost. The absence of pre-computation requires to process a limited amount of data. Indeed the application of the algorithm on a large graph would lead to prohibitive response-times. Nonetheless the sub-graph imported at each query should be small. First a consequent proportion of the online SPARQL endpoints limits the amount of triples retrievable using result-size thresholds: *44.44 percent of the available endpoints are suspected to enforce a result-size thresholds and 10.000 is the most common result-size threshold* [11]. Second the transfer cost should be low in order to avoid a degradation of the response-time. There is an important interweaving between the query-time and remote data requirements.

6.2.2 Software architecture

We present now our solution to meet the requirements. We propose to *locate the processing on the graph* in order to avoid prohibitive response-times. We shift the implementation problem from the processing of a massive amount of data to the smart selection of a limited sub-graph per query. In other words the algorithm will be applied on a small data subset that is sufficient to solve the information need. However there is always the need to *draw meaningful boundaries* over linked data graphs to identify a result set. In our case the challenge resides specifically in the fact that the method has to be applicable at query-time and on remote data. For this we rely mainly on two interdependent features:

- The algorithms are designed to leverage the semantics in order to filter (class-propagation domain) and prioritize the data (similarity measure) reached by the propagation. In other words the semantics of the nodes determine the level of activation in the paths followed by the propagation.
- We couple the spreading activation algorithm to an import procedure that replicates at query-time a sub-graph in a local and transient triple store. The neighbors of the most activated nodes' are imported along the iterations until a limit of triples is reached.

To sum up, our approach is to locally apply the algorithm on a replicated sub-graph that is expanded in accordance with the activation values. In other words there is a resonance between the spreading activation and the graph imported. The main steps of the processing of query are presented below:

[11] http://sparqles.okfn.org/performance

- First a local Kgram[12] instance is created. Kgram is a semantic web framework that combines a triple store and a SPARQL endpoint. This instance imports the neighborhood of the node(s) of interest filtered with the class propagation domain. The import is performed thanks to a local query using a service operator.
- As the propagation spreads along the iterations, the neighborhoods of the most activated nodes are imported until a limit of triples is reached.
- The propagation stops after a specified number of iterations. The most activated nodes constitute then the result set. The results are ranked by decreasing order of activation.

The import operation is illustrated on Figure 6.1. This illustration shows the beginning of the third iteration: the neighbors of *Gustave Courbet*, surrounded by a pink halo, are about to be imported. The neighbors of *Alfred Sisley* and *Édouard Manet* were already imported during the second iteration. The nodes' colors symbolize their activation values. Details about the functioning for both the monocentric and polycentric queries are given in the following subsections.

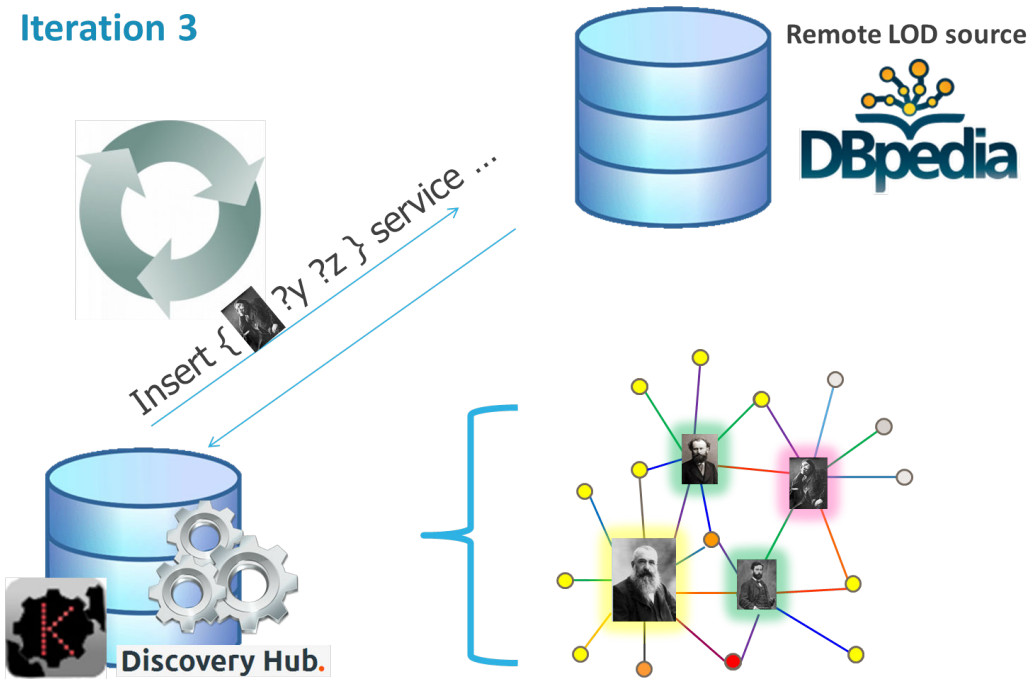

Figure 6.1: Illustration of the incremental graph importation, coupled with the spreading activation process

[12]https://wimmics.inria.fr/corese

6.2.3 Dataset

The implementation was built on top of DBpedia. First DBpedia offers the best generic knowledge coverage of the LOD due to its encyclopaedic nature. Second it is cross-domain and captures a very heterogeneous knowledge in a single graph. It is consequently adapted to our objective of solving composite, potentially heterogeneous, interest queries. Third it can support users' experiments as it contains numerous common-knowledge topics such as films or musical bands. Using it also provides an interesting comparison basis since it is extensively used the research community, see chapter 4. Fourth DBpedia is a more and more mature dataset: its semantics quality and its size increase along the versions[13]. Yahoo notably used some DBpedia extraction components to build its knowledge graph [20]. We did not combine several DBpedia chapters because such combination raises processing problems that are out of the scope of this thesis. The major problem is that the semantic and structural differences of the knowledge sources might create a distortion when a connectionist algorithm liked spreading activation is applied. The most connected and semantically described source will probably be over-represented in the results. Sophisticated balancing strategies need to be researched.

The version we use is the 3.7. As we needed to query the SPARQL endpoint million times during our analysis, we set up a local version. We needed to control the quality and the continuity of service. However the version we set up was remotely accessed by the code through *service* SPARQL queries, similarly to an external and public SPARQL endpoint. The version we set up contains the *wikiPageWikilink* triples[14]. This property indicates that a hypertext link exists in Wikipedia between 2 pages (resources) but that the semantics of the relation was not captured. It provides an amount of extra-links that are very valuable for connectionist methods such as spreading activation. The choice of the DBpdia 3.7 version was initially motivated by an evaluation presented in chapter 8, see section 8.2 page 204. During this experimentation we compared our algorithm results against the results of a system using the DBpedia 3.7 version. It was not updated it in order to maintain the consistency of the analysis performed all along the thesis.

DBpedia is a very dense and heterogeneous dataset. The appendix A on page 233 presents a set of graph metrics of the DBpedia 3.6 graph[15], which is close to the 3.7 one. Some characteristics[16] of DBpedia 3.7 are presented hereafter:

- **Graph size**: 3.64 million resources, 270 million triples.
- **Ontology**: 320 ontology classes, 750 object properties, 893 datatype properties.
- **Typing**: 1.83 million of resources (more than 50 percent) are classified in a

[13] http://blog.dbpedia.org/
[14] http://dbpedia.org/ontology/wikiPageWikiLink
[15] http://blog.dbpedia.org/2011/01/17/dbpedia-36-released/
[16] http://blog.dbpedia.org/2011/09/11/dbpedia-37-released-including-15-localized-editions/

consistent ontology, including 416.000 persons, 526.000 places, 106.000 music albums, 60.000 films, 17.500 video games, 169.000 organizations, 183.000 species.

- **Average degree** of the DBpedia resources[17]: 40.

Today 15 DBpedia chapters are accessible online[18]. As already mentioned in chapter 3 their knowledge overlap to a certain extent[19] but they vary significantly in what they describe and how it is described. 5 DBpedia chapters propose more than 100 millions triples: the English, French, German, Italian and Spanish ones. The English-speaking DBpedia chapter is the main dataset we use, it is simply mentioned as *DBpedia* in this thesis.

6.2.4 Settings

We set up the following variables in order to implement our formulas:

- The propagation spreads in both directions to take into account the incoming and outcoming linked to the neighbors. From a spreading activation point of view the orientation is arbitrary and depends on a modeling choice.
- The *threshold* filtering the propagation domain is experimentally set to a low value of 0.01. In other words the classes having a prevalence inferior to 1% are excluded from the $CPD(o)$. We do not want to filter too much the class propagation domain in our exploratory search context. As mentioned by [147] the indirect links between the resources bring useful knowledge such as elements of context. Nevertheless a very low threshold can exclude uninformative types and lower the processing complexity.
- In DBpedia, the instances are linked to their categories through the *dcterms:subject*[20] property. We use this property to compute the $commontriple(i, o)$ function i.e. $p = dcterms : subject$. The categories constitute an orthogonal topic taxonomy which is very informative about the resources. They expose some of their salient aspects. The value of using the DBpedia categories to enhance recommendations was shown in [43]. Moreover 99.42% of the resources have categories associated to them [108]. The main reason of using only the *dcterms:subject* property to compute $commontriple(i, o)$ is that the similarity has to be very cheap to compute. A large number of similarities have to be computed on run-time during a query. Moreover the framework results combine similarity and relatedness, so element of knowledge that are not taken into account for the similarity computation will be processed by the spreading activation procedure in a *structural* way.

[17] http://dbpedia.org/resource/
[18] wiki.dbpedia.org/Internationalization/Chapters?v=190k
[19] dbpedia.org/Datasets39/CrossLanguageOverlapStatistics
[20] http://purl.org/dc/terms/subject

- 50% of the DBpedia resources is untyped. We can not exclude this half of the dataset. We consider it as a mass of knowledge that has not been typed yet. Consequently in our implementation the untyped nodes are associated to a fictive *Miscellaneous* type and are processed by the spreading activation. Such *Miscellaneous* nodes appear in the result set of the Discovery Hub web application.

- Due to the inconsistencies present in the DBpedia data the inference feature of Kgram was deactivated for the automatic deduction of instances types. Making automatic inferences on DBpedia can lead to wrong results [190] and exposing them to the users might lead to a loss of confidence in the system.

The maximum number of iterations and the size of the sub-graph imported have still to be set and are discussed in the following sections.

6.3 Monocentric queries implementation

In the case of monocentric queries the triples importation follows directly the spreading activation logic. The graph is loaded iteratively regarding the nodes activation values, until the limit is reached.

6.3.1 Code and main SPARQL queries

The pseudo-code for a monocentric spreading activation is presented hereafter. The line 2 corresponds to the initialization of the algorithm: the class propagation domain of the node of interest is computed and its neighbors are locally imported. The lines 6 to 14 correspond to the computation of the activation level for each node at each iteration. The lines 16 to 22 correspond to the import process coupled to the spreading activation algorithm. The non-trivial queries used to execute the algorithm are also presented below. The query 6.1 is used to identify and count the occurrences of the types of the query-node neighbors. The line 4 aims to retrieve the outcoming neighbors and their types, the line 6 do the same for the incoming neighbors. We specify that we are only interested in the DBpedia ontology types with a $filter()$ clause, line 8. We also discard uninformative resources i.e. resources that are linked to the topic of interest through disambiguation or redirection properties (line 8 and 9). Then the deepest types occurrences are identified by deduction. For instance, suppose that we have *Artist, Person and Writer* instances. *MusicalArtist* and *Writer* are sub-classes of the *Person* class. Thus we can subtract the total of *MusicalArtist* and *Writer* to the *Persons* total in order to obtain the amount of *Person* instances for which *Person* is the deepest type available. To this end we use the *depth* function of Kgram. We start from the deepest classes in

the branches of the ontology concerned by the currently processed query.

```
   Data: o ∈ KB, maxPulse ∈ ℕ, limitImport ∈ ℕ
   Result: activationHashMap(i,a); i ∈ KB; a ∈ ℝ
 1 begin
 2     activationHashMap.put(o, 1.0);
 3     getClassPropagationDomain(o);
 4     importNeighborhood(o);
 5     while i ≤ maxPulse do
 6         foreach i in activationHashMap do
 7             if activationHashMap.get(i) > 0 then
 8                 for j in neighbor(i) do
 9                     sim = commontriple(j, o);
10                     act = activationHashMap.get(i) * (1 + sim)/degree(i);
11                     tempMap.put(j, tempMap.get(j) + act);
12                 end
13             end
14         end
15         activationHashMap = tempMap
16         foreach k in sortedByDecreasingValue(activationHashMap) do
17             if importSize >= limitImport then
18                 break();
19             else
20                 importNeighborhood(k);
21             end
22         end
23         i++;
24     end
25 end
```

Algorithm 1: Monocentric semantic spreading activation pseudo-code

The import is split up in several queries in order to speed up the processing as the SPARQL endpoints have generally the capability to process several queries in parallel. The queries 6.2 and 6.3 are sent to the targeted SPARQL endpoint by the Java code when the local graph is expanded. The example queries 6.2 correspond to the import of the outcoming nodes at the iteration 1 for the query *Claude Monet* (a similar query is performed to retrieve the incoming ones). The line 5 specifies that we are interested in the outcoming nodes. Again several uninformative properties are discarded (redirection, disambigutation) as propagating the activation through them is not relevant; see the lines 7 to 10. The line 13 and 14 shows that the import of the types neighbors is limited to the ones having at least one type in $CPD(o)$.

The line 11 specifies that the untyped neighbors should also be retrieved.

The query 6.3 corresponds to the import of the categories. These categories are used later during the similarity computation. In this example the categories are loaded at the second iteration when the graph is expanded with the neighborhoods of *Gustave Caillebotte*, *Blanche Hoschedé-Monet*, *Alfred Sisley*, *Pierre-Auguste Renoir* and more. The line 5 to 8 shows the filter condition used to retrieve only the categories associated to Claude Monet. The lines 9 to 14 shows that we retrieve these categories for the outcoming neighbors of the nodes being currently imported (*Gustave Caillebotte*, etc.). We do the same for the incoming ones in another bloc, see lines 16 to 20.

```
1  select * where {
2   service <sparqlendpointurl> {
3    select distinct ?t (count(?t) as ?tcount) where {
4    {?x ?y ?z . ?z rdf:type ?t}
5    UNION
6    {?z ?y ?x . ?z rdf:type ?t}
7    filter (?x=<dbpedia:Claude_Monet>)
8    filter(regex(?t,"http://dbpedia.org/ontology"))
9    filter(?y!=<dbpedia-owl:wikiPageRedirects>
10   && ?y!=<dbpedia-owl:wikiPageDisambiguates>)
11   }
12 group by ?t }
13 }
```

Listing 6.1: SPARQL query identifying and counting the occurrences of the types of query-node neighbors

```
1  INSERT {?x ?p1 ?y1 .
2  ?y1 rdf:type ?k1 }
3  where {service <sparqlendpointurl> {
4   select ?x ?p1 ?k1 ?y1 where {
5    ?x ?p1 ?y1
6    filter(!isLiteral(?y1) && ?p1!=rdf:type
7    && ?p1!=owl:sameAs
8    && ?p1!=<dbpedia-owl:wikiPageInterLanguageLink>
9    && ?p1!=<dbpedia-owl:wikiPageRedirects>
10   && ?p1!=<dbpedia-owl:wikiPageDisambiguates>)
11   OPTIONAL  {?y1 rdf:type ?k1}
12   filter( ?x = <dbpedia:Claude_Monet> )
13   filter(?k1=<dbpedia-owl:Museum>
14   || ?k1=<dbpedia-owl:Writer> || ... )
15 }}}
```

Listing 6.2: SPARQL query for the import of the outcoming neighborhood of Claude Monet filtered by the class propagation domain

```
1  INSERT {?y <http://purl.org/dc/terms/subject> ?prop }
2  where {sparqlendpointurl> {
3   select distinct ?y ?prop where {
4    ?y <http://purl.org/dc/terms/subject> ?prop
5    filter(?prop = <dbpedia:Category:Impressionists>
6    || ?prop = <dbpedia:Category:People_from_Paris>
7    || ?prop = <dbpedia:Category:French_Painters>
8          ....)
9    {?x ?p ?y
10   filter( ?x = <dbpedia:Gustave_Caillebotte>
11   || ?x = <dbpedia:Blanche_Hoschede_Monet>
12   || ?x = <dbpedia:Alfred_Sisley>
13   || ?x = <dbpedia:Pierre-Auguste_Renoir> )
14   }
15   UNION {
16    {?y ?p ?x
17    filter( ?x = <dbpedia:Gustave_Caillebotte>
18    || ?x = <dbpedia:Blanche_Hoschede_Monet>
19    || ?x = <dbpedia:Alfred_Sisley>
20    || ?x = <dbpedia:Pierre-Auguste_Renoir> )
21 }}}}}
```

Listing 6.3: SPARQL query for the import of the categories used for the similarity measure computation

6.3.2 Algorithm behavior analysis

The maximum number of iterations and the limit of triples processed per query are two critical settings for the implementation. They are major variables influencing the tradeoff between the quality of the results and the computational cost. We needed to deeply understand the algorithm behavior in order to set them. We enumerate below several hypothesis we wanted to verify as well as the analyses we performed for this purpose.

- **Hypothesis 1: Convergence**. The algorithm converges quickly, in few iterations.
- **Hypothesis 2: Import size**: Running the algorithm over a carefully selected amount of triples constitutes a sufficient approximation.

- **Hypothesis 3: Distance**. The algorithm retrieves distant nodes in the top results i.e. nodes that are not part of the query-node direct neighborhood.

We do not have a golden truth at this point. Thus we consider that the best results possible are the stable ones, after the algorithm converged, and with the largest import. The chapter 8 is dedicated to the perceived relevance of the results by the users. The current chapter focuses on how to implement the algorithm with the best trade-off between the response-time and the results alteration.

6.3.2.1 Method and hypotheses

In order to reduce the cost of the analyses we performed them on a subset of 100.000 queries. To select such query-resources we referred to the sampling methods aiming to select a representative subgraph. According to [110] the best sampling method to preserve large graph properties is a random walk. We followed this recommendation and computed a 100.000 resources DBpedia sample using this method. Both the random walker code and the sample itself are accessible online[21].

To compare the result lists we obtained with various configurations we notably used the Kendall's Tau τ_b rank correlation coefficient [92]. τ_b is a measure reflecting the concordance of two ranked lists where -1 corresponds to a total discordance and 1 corresponds to a total concordance of the ranks. The τ_b formalization is presented in the appendix B on page 235.

Our hardware configuration for all the analyses presented in this chapter was:

- Application server: 8 proc Intel Xeon CPU E5540 @2.53GHz 48 Go RAM
- SPARQL endpoint: 2 cores Intel Xeon CPU X7550 @2.00GHz 16Go RAM

6.3.2.2 Convergence

We chose to use a maximum number of iterations as the stop condition. It has the advantage of limiting the response-time relatively uniformly for all the queries. In order to determine the best number of iterations in our context we observed the algorithm convergence. We performed an analysis using the 100.000 queries. First we counted the number of top 100 shared results between the $n-1$ and n iterations. Then we computed the τ_b coefficient on these shared results in order to have an indication on the ranking convergence. We studied the first hundred iterations. The triples loading limit is not studied yet and is experimentally set to 10.000 for this first analysis. The value of 10.000 corresponds to the most common limit of retrievable triples used by the online SPARQL endpoints[22].

For clarity purpose the figures 6.2 and 6.3 show only the first twenty iterations. Indeed we observe that the top results are quickly converging: after 16 iterations

[21] http://semreco.inria.fr/hub/tools
[22] http://sparqles.okfn.org/performance

the percentage of average top 100 shared results exceeds 99% (figure 6.2) and the average τ_b is also superior to 0.99. It is clear that the results change very slowly after few iterations. In others words it becomes very expensive to continue the processing after few iterations considering the very slow evolution of the results. The algorithm converges quickly, thus the hypothesis 1 is verified. We decided to set the maximum number of iterations to 6 because the curves are very flat after 6. The relation between the convergence and the linked dataset metrics is discussed in the section 6.6 of this chapter.

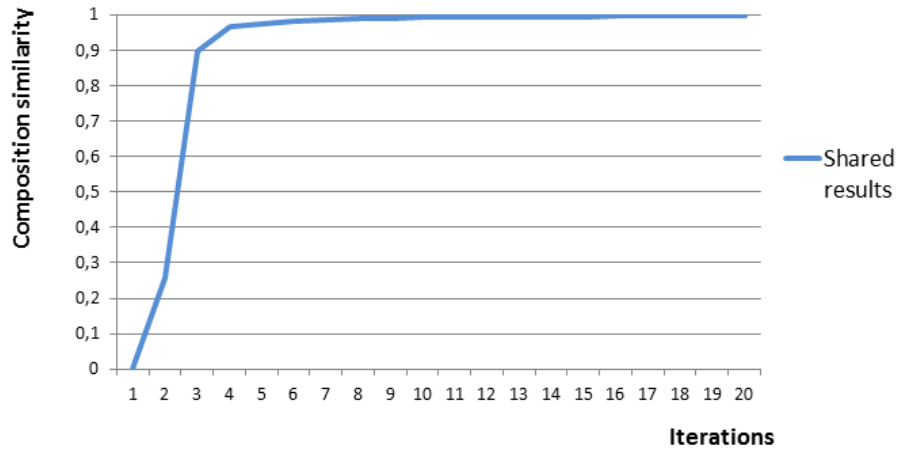

Figure 6.2: Average top 100 shared results among iterations $n-1$ and n, monocentric queries

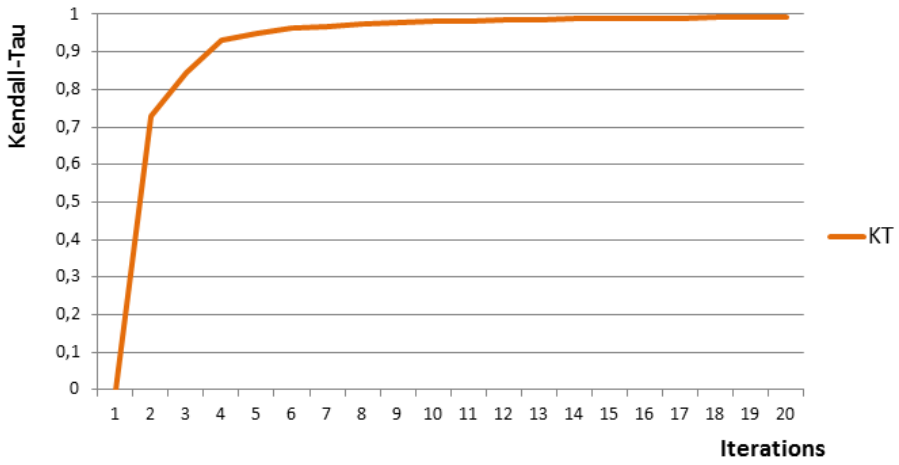

Figure 6.3: Average τ_b between the top 100 shared results at iterations $n-1$ and n, monocentric queries

6.3.2.3 Import size

In order to control the size of the sub-graph processed and by extension the response-time we introduced a maximum limit of triples imported per query. Indeed as the DBpedia network is dense a very high amount of nodes can be activated in few iterations, even with the $CPD(o)$ filtering operation.

This limit is the second parameter that required extensive analysis. Indeed, processing a sub-graph only can be considered as an approximation. We need to observe if the results are not too much altered regarding the amount of triples processed. We used again the 100.000 analysis queries. Each query was executed ten times with triples import limits ranging from 2000 to 20.000 triples, with a step of 2000. The maximum is set to 20.000 because the transfer cost and the data provider restrictions make it impossible the consumption of large amount of remote data. Nonetheless, as mentioned previously the interest of the method is to consume a small amount of data on demand. Thus the objective of the upcoming analysis is to determine the smallest size tolerable for the import. The figure 6.4 shows first that the algorithm response-time is linear with regard to the triples loading limit.

The figure 6.5 shows the amount of shared top 100 results, at the end of the 6 iterations, from a loading limit to another (2000 by 2000). The figure 6.6 shows the corresponding τ_b. It is observable that for the imports superior to 6000 triples the changes are very small. Knowing that the response-time is linear it is very expensive to augment the size of the import after 6000 considering the minor results changes it brings. In other words, an import 6000 triples gives a good trade-off between performance and results alteration. The hypothesis 2 is validated.

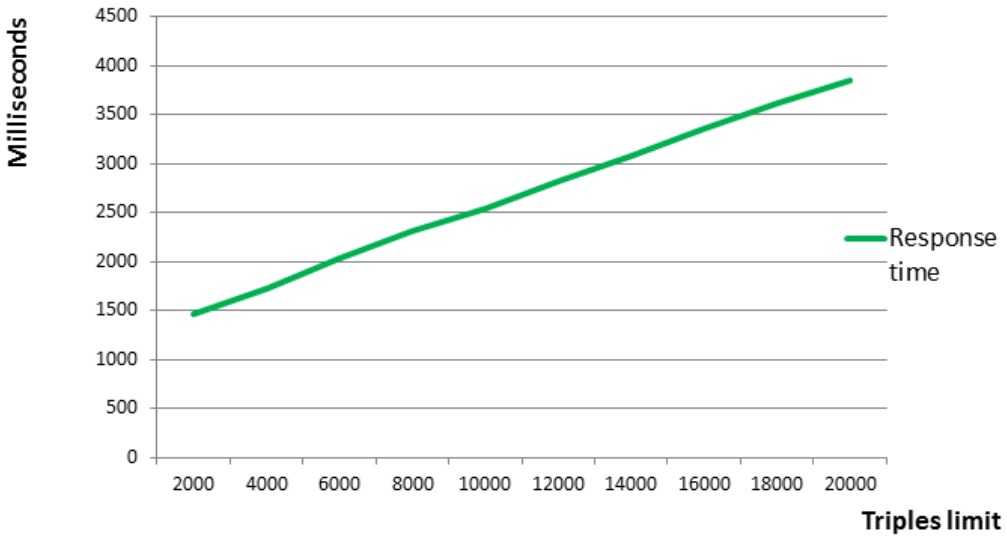

Figure 6.4: Monocentric queries response-time against increasing triples loading limits

The figure 6.7 shows the distribution of the triples importation over the iterations with a limit of 6000. It is observable that without this limit the amount

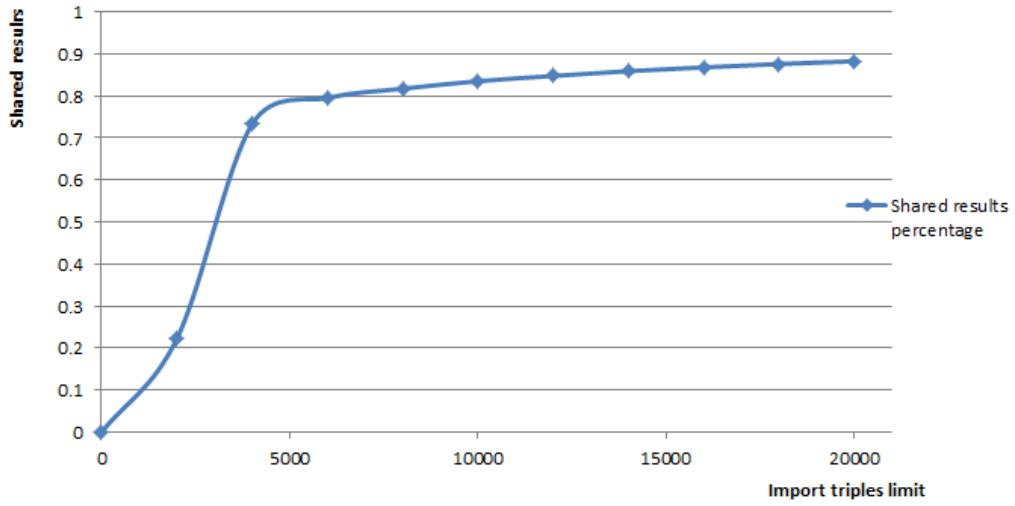

Figure 6.5: Top 100 shared results from one loading limit to another, by increment of 2000 triples

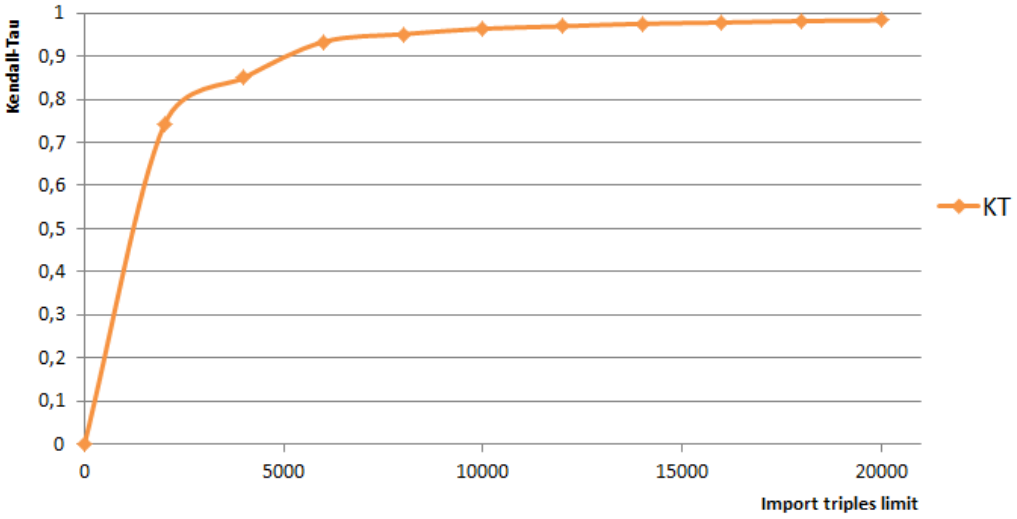

Figure 6.6: τ_b between the top 100 shared results from one loading limit to another, step of 2000 triples

of triples imported would explode. This figure also illustrates an interest of the $commontriples(i,o)$ similarity measure: as the majority of the triples are loaded during the second iteration there is a need to orient quickly the propagation toward relevant resources. The $commontriples(i,o)$ functionality assumes notably this role.

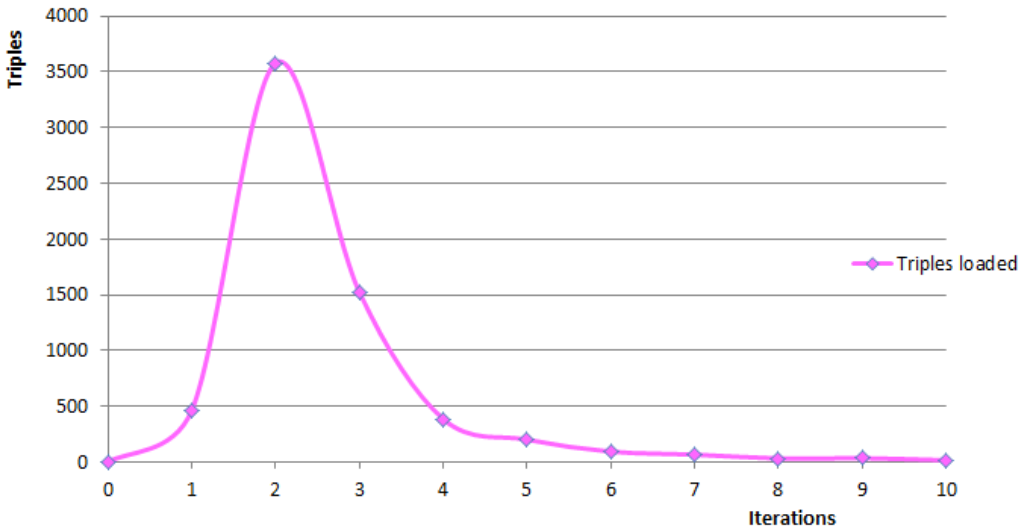

Figure 6.7: Average amount of triples imported at each iteration, monocentric query

6.3.2.4 Distance

One interest of the traversal algorithms is that they are able to retrieve *distant* results i.e. nodes that are not direct neighbors of the query-node. In other words the algorithm is able to expose relations that do not exist as direct links in the original graph. The figure 6.8 shows the distance (shorted path) of the top 10 and top 100 results with the query-node. It also shows the average maximum distance in the top 100 results. As these analyses were done later during the thesis, they were also executed on the French and Italian SPARQL endpoints for comparison purpose. For the English-speaking chapter it is observable that indirect neighbors are present in the top 10 and in the top 100 results (average distance superior to 1) and that the maximum distance is over 2. This last point means that it is not unusual to retrieve 3-hop nodes in the results. Thus the hypothesis 3 is verified. The scores are even higher for the French and the Italian endpoints. They are less dense than the English-one, thus the spreading activation is more likely to reach distant nodes.

6.3.2.5 Performance

The figure 6.9 shows the distribution of the 100.000 queries ordered by their response-time. The parameters previously discussed were used during the analysis i.e. the maximum number of iterations set to 6 and the triples loading limit set to 6000. It is observable that these parameters limit the response-time relatively uniformly. The nodes on the left are very connected ones e.g. countries. The response-time is 2.3 seconds on average.

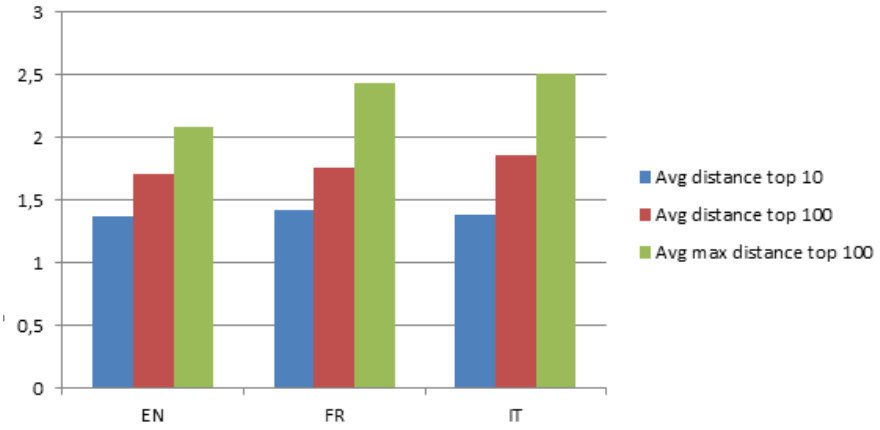

Figure 6.8: Average result distances: top 10, top 100, maximum in top 100

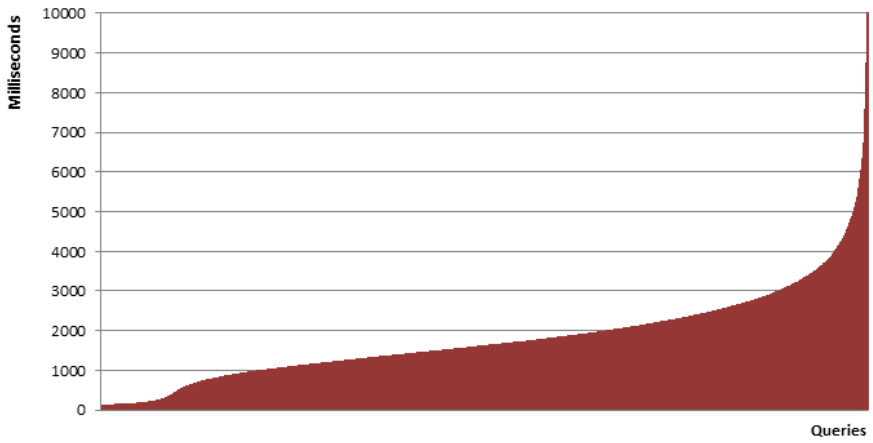

Figure 6.9: Distribution of the monocentric queries ordered by their response-time, with the triples loading limit at 6000 and the maximum number of iterations at 6

6.4 Polycentric queries implementation

Identifying and exploring the linked data knowledge at the crossroad of several topics is very difficult and requires a solid support for the users. An efficient implementation of the polycentric queries is important in the context of exploratory search where they can help unveil hidden relations between several topics of interest. There is a specific difficulty when implementing such queries in the context of large and dense graphs of the LOD. Indeed, in the case of monocentric queries the propagation spreads *around* a single node. On the contrary in the case of polycentric queries, there is a need to drive several unitary propagations in a shared zone that is at the cross-road of the stimulated nodes. It would be very computationally expensive to use the monocentric sub-graph import approach from all the stimulated nodes and to wait for an eventual overlap of the propagations to hap-

pen. Instead of that we rely on the identification of paths between the stimulated nodes.

6.4.1 Code and main SPARQL queries

First the shared propagation domain is computed, as formalized in chapter 5. Then we figure if the stimulated nodes are direct neighbors. If it is the case their neighborhoods are loaded and the propagation starts. If it not the case a two-arc non-oriented SPARQL path query is sent to the endpoint in order to identify one or several paths between them. We use the Kgram *path*() function[23] for this purpose, see the query 6.4 below. The Kgram *path*() function is not standard. It allows to easily declare sequences of properties thanks to proprietary commands[24]. In case the unoriented query does not produce any result or if the SPARQL endpoint refuses it because it is too complex we search oriented paths between the seeds in one direction, then in the other one. We do it for a length of 3 or 4 if necessary. If at this point we are still unable to find a path between the stimulated nodes the query fails. However in this case the nodes are very distant and the results retrieved would be probably poorly relevant and very expensive to produce. Searching for oriented paths is not consistent with the spreading activation algorithm that spreads in both directions. However it is a useful approximation for the queries combining distant nodes because it lowers the complexity of the query and allows retrieving paths that are not identifiable with unoriented querying.

The query 6.4 is an example of unoriented path query using the Kgram *path* function, see the lines 4 to 10. The circumflex symbol means that the path is undirected and the *kg:path kg:expand 2* command on line 14 means that the paths we want to retrieve as a maximum lenght of 2. As it can be observed on the query 6.4 that only the *wikiPageWikiLink* properties, which are the most prevalent in our DBpedia implementation, are used for this path identification. The *wikiPageWikiLink* properties capture a very high number of connections between the DBpedia resources. Moreover when two nodes are linked by a semantically-defined property (e.g. *http://dbpedia.org/ontology/memberOf*) the relation is often mentioned in the Wikipedia plain text. Consequently a corresponding *wikiPageWikiLink* triple is also generated. Thus, restraining the path queries to these properties leads to no knowledge loss in our context. The nodes' neighborhoods that are found after the path identification procedure are then loaded in the local Kgram instance by increasing degree order. We assume that nodes having a lower degree are more informative about the connections between the stimulated nodes. To maximize the chance of retrieving results the pivot nodes identified by the SPARQL path queries

[23] https://wimmics.inria.fr/node/37

[24] e.g. retrieving the nodes that are linked though an unoriented sequence of 3 properties *dbpedia-owl:wikiPageWikiLink*

are eligible for activation even if they do not have a type present in $CPD(O)$.

```
Data: O = {o_1, o_2, ...o_n} ; o_n ∈ KB, maxPulse ∈ ℕ, limitImport ∈ ℕ
Result: polyActivationHashMap(i,a); i ∈ KB; a ∈ ℝ
1  begin
2      foreach o_n in O do
3          activationHashMap_n.put(o_n, 1.0)
4      end
5      getSharedClassPropagationDomain(O)
6      importPath(getPathBetweenSeeds(O))
7      while i <= maxPulse do
8          foreach o_n in O do
9              foreach i in activationHashMap_n do
10                 if activationHashMap.get(i) > 0 then
11                     for j in neighbor(i) do
12                         sim = commontriple(j, o_n);
13                         act = activationHashMap_n.get(i) * (1 + sim) / degree(i);
14                         tempMap.put(j, tempMap.get(j) + act);
15                     end
16                 end
17             end
18             activationHashMap_n = tempMap ; tempMap.clear() ;
19         end
20         foreach i in activationHashMap_1 do
21             temp = 1
22             foreach o_n in O do
23                 if activationHashMap_n.containsKey(i) then
24                     temp = activationHashMap_n.get(i) * temp
25                 else
26                     temp = 0 ;
27                 end
28             end
29         end
30         polyActivationHashMap.put(i, temp / log(degree_i))
31         foreach k in sortByIncreasingDegree(polyActivationHashMap) do
32             if importSize => limitImport then
33                 break()
34             else
35                 importNeighborhood(k)
36             end
37         end
38         i++
39     end
40 end
```

Algorithm 2: Polycentric semantic spreading activation pseudo-code

It should be mentioned that for the polycentric queries combining more than 2 resources the paths for all the pairs should be identified. It is computationally expensive. This is the reason why the Discovery Hub application only allows polycentric queries combining 4 nodes at the maximum.

Once the paths are imported several unitary propagations starts from each stimulated node. At each iteration a polycentric activation value for each node is computed according to the definition 13 of the chapter 5. At each iteration, if the import limit is not reached the most activated (at a polycentric level) nodes neighborhood are imported in order to pursue the propagation in relevant parts of the graph. The import triples limit is multiplied by the number of nodes originally stimulated e.g. 12.000 for a composite query combining 2 resources.

The pseudo-code of the polycentric queries is presented here-after. The lines 2 to 6 correspond to the initialization phase. First the shared class propagation domain is computed. Then the paths between the nodes of interest are identified thanks to the SPARQL queries previously mentioned. The nodes composing these paths (and their neighbors are locally imported). The lines 8 to 20 show the activation values computation for each of monocentric propagations. The lines 21 to 30 correspond to the computation of the activation values at a polycentric level. The lines 32 to 38 show the triples importation process coupled to the polycentric activation values.

```
 1 SELECT ?x WHERE {
 2 service <sparqlendpointurl> {
 3  select distinct ?x ?degree where {
 4  <dbpedia:The_Beatles>(<dbpedia-owl:wikiPageWikiLink> |
 5  ^(<dbpedia-owl:wikiPageWikiLink>)) + ?x
 6  { ?x <dbpedia-owl:wikiPageWikiLink>
 7  <dbpedia:Ken_Loach> }
 8  UNION
 9  {<dbpedia:Ken_Loach>
10  <dbpedia-owl:wikiPageWikiLink>  ?x }
11  filter(!isLiteral(?x)) }
12  }
13 }
14 pragma {kg:path kg:expand 2}
```

Listing 6.4: Polycentric path detection with an unoriented query

6.4.2 Algorithm behavior analysis

6.4.2.1 Method and hypothesis

Like for monocentric queries we need to study the convergence and the response-time of the polycentic algorithm. Several questions that are specific to this form of querying also emerge. We formulated the following hypotheses:

- **Hypothesis 4: path identification**. It is possible to find paths for a majority of instance combinations.
- **Hypothesis 5: convergence**. The algorithm converges quickly i.e. in few iterations.
- **Hypothesis 6: response-time**. The limit of triples imported helps to control the response-time.
- **Hypothesis 7: result set specificity**. The results are specific to the polycentric queries; they are at the crossroad of resources of interest.
- **Hypothesis 8: results' distances**. The results are equally distant from the query-resources.

The hypothesis 4 was verified thanks to a set of polycentric queries generated from the Discovery Hub users' Facebook *likes*. In order to verify the hypotheses 5, 6, 7 and 8 we reused the 100.000 nodes obtained from a DBpedia sampling. For each of them we selected randomly 2 nodes in the 2 arc-max neighborhood thanks to a random walker. We then processed 3 queries: one with the sample node only (monocentric) and two polycentric queries by combining it with each of its randomly selected neighbors. We selected 2 nodes in the 2 arc-neighborhood because DBpedia is very dense. Therefore the probability that nodes composing a composite query are close in terms of shortest path is high. This point is confirmed by the analysis presented hereafter.

6.4.2.2 Path identification

We relied on the users' Facebook *likes* imported in Discovery Hub to verify hypothesis 4. For each user having its *likes* imported in the web application we created all the possible combinations of 2 resources i.e. all the possible bi-centric queries corresponding to two of their likes which led to a set of 700 realistic queries. The *likes* were matched thanks to SPARQL queries that aimed to find a correspondence between their labels and a DBpedia resource one. Then the matches were manually checked. At the end approximately 60% of the likes were successfully matched to DBpedia resources. The objective was to simulate real potential composite interest queries. For each combination (e.g. *Simon Garfunkel* and *Aldous Huxley*) we stored the type of query that successfully retrieved a path i.e. unoriented at distance 1, 2 or oriented at distance 3, 4. The table 6.1 below presents the proportion of paths that were found regarding the type of queries. It also shows the average complete algorithm response-time for each type of queries. It is observable that 94.72% of the paths identifications were successful. The majority of the them was found using the 2-arc unoriented query (84.13%). As mentioned before the DBpedia graph is very dense and the shortest path between the query-nodes is often short. Nevertheless the oriented queries helped to find paths for approximately 10% of the combinations but at the cost of a very high response-time. 5.28% of the paths identification failed. It corresponds to cases where the composite query

seeds are very distant. It is very expensive to identify a path between them and it results in SPARQL endpoint timeouts.

Query	Unoriented 1	Unoriented 2	Oriented 3	Oriented 4	Failed
Amount	202	387	64	10	37
Percentage	28.85	55.28	9.14	1.42	5.28
Response-time in ms	4396	4241	26727	40404	N/A

Table 6.1: Proportion of queries regarding the path identification method and their average response-time

If we refer to this analysis we can affirm that the hypothesis 4 is verified as 94.72% of the paths identifications were successful.

6.4.2.3 Convergence analysis

As the activation level of a polycentric query is the product intersection of the monocentric queries propagation (attenuated by the nodes' degree) the convergence should be similar to the monocentric ones. We verified it by comparing the amount of top 100 shared results and the corresponding τ_b from one iteration to another for the 200.000 composite queries. The results are presented on the figures 6.10 and 6.11.

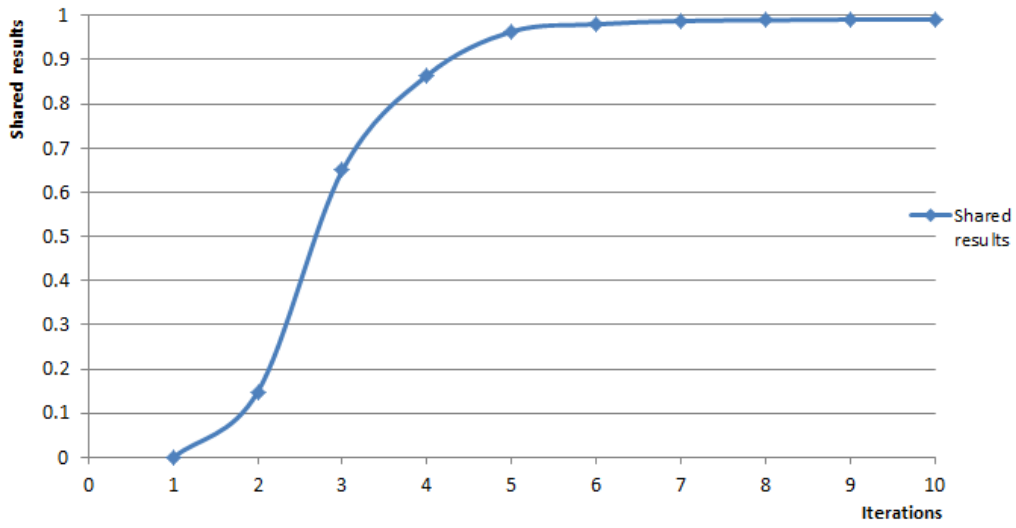

Figure 6.10: Average top 100 shared results among iterations $n-1$ and n, polycentric queries

We observe that, on average, the result lists are very stable after 6 iterations, both in terms of composition and ranking. It is not a surprise as it has been shown previously that the mono-centric queries were converging around 6 iterations also. The hypothesis 5 is verified.

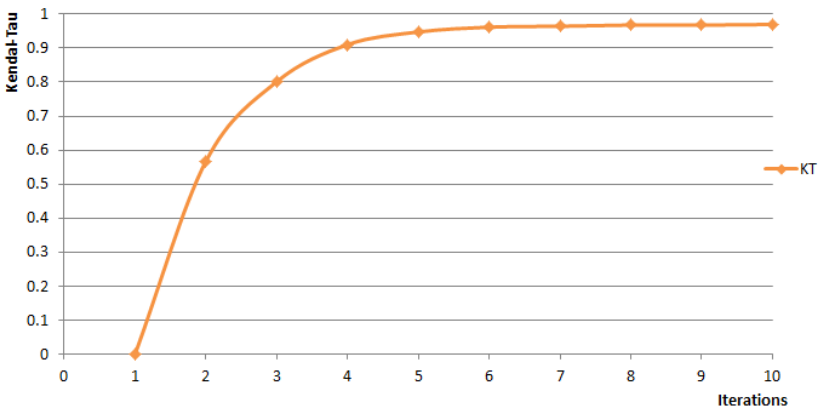

Figure 6.11: Average τ_b between the top 100 shared results at iterations $n-1$ and n, polycentric queries

6.4.2.4 Response-time

Distribution of the polycentric queries ordered by their response-time, shown on the figure 6.12, shows that a majority of queries are processed in few seconds. To be precise the polycentric queries are processed in 4.2 seconds on average. Thus the hypothesis 6 is validated. Overall the response-time of the polycentric queries is superior to the monocentric ones due to the path identification cost and the larger sub-graph imported.

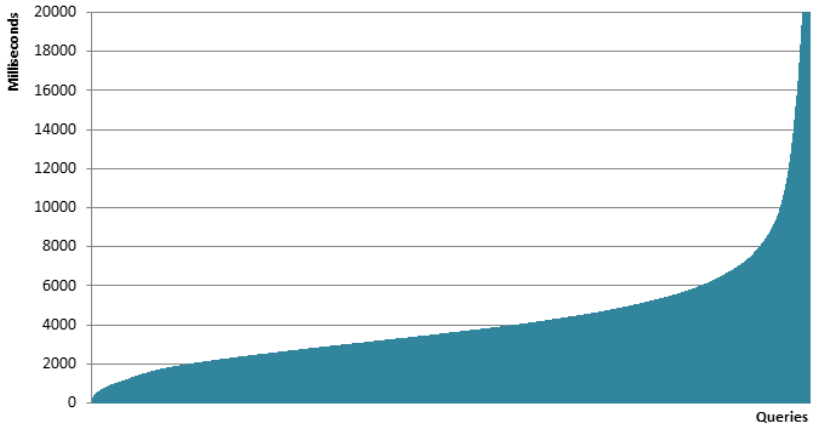

Figure 6.12: Distribution of the polycentric queries ordered by their response-time

The figure 6.13 shows the average amount of triples loaded at each iteration. Contrary to the monocentric queries it is visible that a large amount of triples is loaded at the first iteration. It corresponds to the import of the initial sub-graph which is constituted of the paths between the nodes, and of the neighbors of the nodes that constitute the paths. The 12000 (instead of 6000) triples loading limit is also visible on the figure.

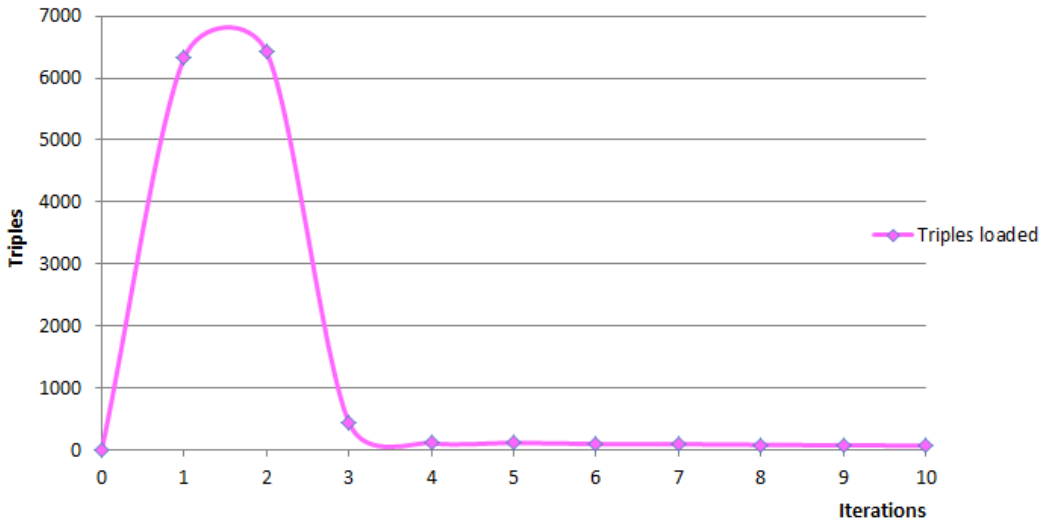

Figure 6.13: Average amount of triples imported at each iteration, polycentric query

6.4.2.5 Results specificity

We wanted to observe if the polycentric results were specific to the inputs combination. For this we compared the results obtained for the 100.000 DBpedia sample monocentric queries and the 2 * 100.000 polycentric queries amongst each other. The objective was to verify that the result set retrieved is well and truly at the crossroad of the query-nodes composing the query. The figure 6.14 shows the distribution of queries ordered by the average top 100 shared results percentage between the monocentric and the polycentric queries i.e. the average amount of top 100 results shared between the monocentric query results (having X as inputs) and the polycentric ones (having X + Y and X + Z as inputs). The figure 6.15 shows the same information for the 2 polycentric queries amongst themselves i.e. the average amount of to 100 shared results between the composite query having X + Y as inputs and the one having X + Z as inputs. These 2 distributions point out that the top result lists are very different in terms of composition. In others words the composite query results are highly specific to their inputs; they are at the crossroad of the seeds. The hypothesis 7 is verified.

6.4.2.6 Distance

During our analyses we also wanted to observe if the results were equidistant from the 2 query-nodes in term of shortest path. Such equidistance gives an indication about the fact the results retrieved are related to the 2 seeds with an equal strength. This is important as the objective is to explore results at their cross-road. The table 6.2 shows the average distance between the results and the 2 query-nodes. More precisely 3 distances are presented: the average one for the top 10 results, the

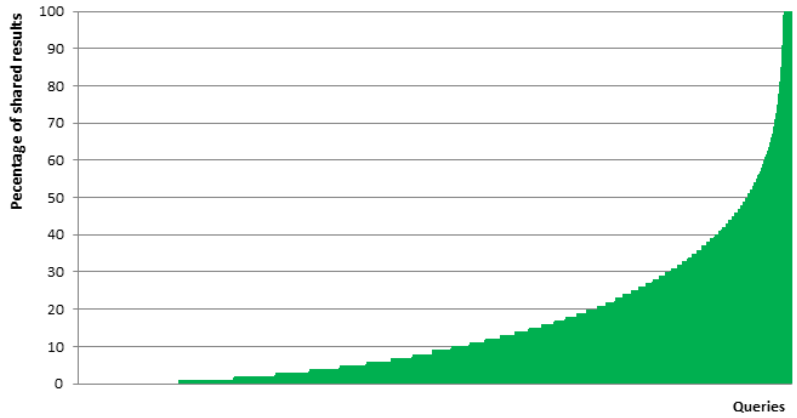

Figure 6.14: Distribution of the queries ordered by the amount of top 100 results between the monocentric and the 2 polycentric queries

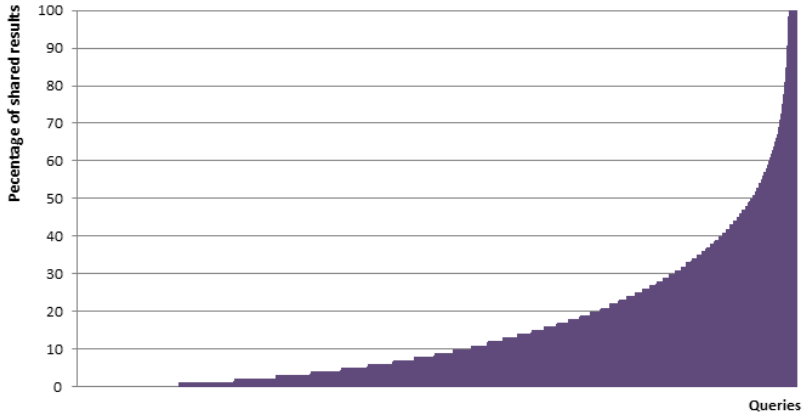

Figure 6.15: Distribution of the queries ordered by the amount of shared top 100 results between the first and second polycentric query results

average one for the top 100 results and the average maximum distance in the top 100 i.e. the furthest node(s) distance in the top 100. It is observable that all these distances are well balanced between the resource 1 and 2. Thus the hypothesis 8 is verified.

6.5 Advanced querying implementation

This section focuses on the implementation of the algorithm variants allowing the users to access topic-centered hardly identifiable knowledge nuances. The idea is to offer deep exploration capacities by allowing exploratory search on topic from multiple angles. A user may enter multiple queries about a topic using diverse perspective operations to get a more complete view about it or to focus on specific facets of interest. We implemented 3 perspective-operations. The first two are the

	Resource 1	Resource 2
Average distance of the top 10 results	1.4	1.5
Variance of the distance of the top 10 results	0.35	0.42
Average distance of the top 100 results	1.42	1.5
Variance of the distance of the top 100 results	0.36	0.46
Average of the maximum distance in the top 100	2.41	2.42
Variance of the maximum distance in the top 100	0.42	0.13

Table 6.2: Average distances and their variances between the results and the 2 query-nodes

criteria of interest specification and the controlled randomness injection presented in chapter 4. The third one is specific to the context of DBpedia and is called the *cultural prism operation*. It consists in deliberately selecting a local DBpedia chapter to explore a topic through a certain culture or geographical location.

6.5.1 Criteria of interest specification

The DBpedia categories are used as criteria of interest/disinterest in our implementation i.e. p = *http://purl.org/dc/terms/subject*. When implementing the criteria of interest specification we chose to keep this unique property for the similarity computation. First a unique property reduces the computational cost. Second it simplifies the interactions for the selection of the criteria of interest (see chapter 7). The resources described in some linked dataset are complex and can be explored through many points of view. We reuse the *Claude Monet* and *Ken Loach* examples in order to show that the DBpedia categories expose a variety of topics facets, see figure 6.16 where we manually grouped them. Such groups of facets inform on the nationality, artistic movement, illness and epoch for Claude Monet. They inform on the artistic aspects, politics, origins and epoch for Ken Loach. The criteria specification variant offers a mechanism to favor or discard such aspects during the results computation.

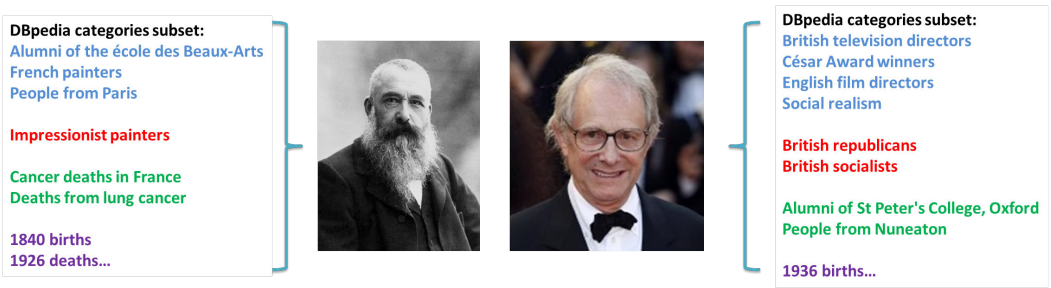

Figure 6.16: Extract of the Claude Monet and Ken Loach DBpedia categories

```
   Data: o ∈ KB, maxPulse ∈ ℕ, limitImport ∈ ℕ
   Result: activationHashMap(i,a); i ∈ KB; a ∈ ℝ
 1 begin
 2     activationHashMap.put(o, 1.0);
 3     getClassPropagationDomain(o);
 4     importNeighborhood(o) ;
 5     while i ≤ maxPulsedo
 6         foreach i in activationHashMap do
 7             if activationHashMap.get(i) > 0 then
 8                 for j in neighbor(i) do
 9                     sim_criteria = commontriple_criteria(j, o, C))
10                     act = activationHashMap.get(i) * (1 + sim_criteria)/degree(i)
11                     tempMap.put(j, tempMap.get(j) + act)
12                 end
13             end
14         end
15         activationHashMap = tempMap
16         foreach k in sortedByDecreasingValue(activationHashMap) do
17             if importSize>=limitImport then
18                 break() ;
19             else
20                 importNeighborhood(k)
21             end
22         end
23         i++
24     end
25 end
```

Algorithm 3: Monocentric semantic spreading activation pseudo-code, criteria of interest specification variant

With the criteria specification variant such categories are associated to a weight assigned by the users before they launch the query. They are consequently finely processed by the algorithm, according to the user tastes and interests. They are not used in an undifferentiated manner as in the basic algorithm. Through the actual interface the users have the choice between three values for each criteria c: not interesting ($v = -1$), neutral ($v = 0$) and interesting ($v = 1$). The functioning of the algorithm is modified as follows (also available with polycentric queries), the similarity used to compute the activation values is changed, see the lines 9 and 10.

By declaring such criteria a user can perform a query specifying that he is interested in Claude Monet because he is an impressionist but not because he is French. Examples of such queries and the top results they retrieve are presented in the table 6.3. The results presented in the table are the top *Artist* facet results. The DBpedia categories often reflect pieces of information associated to a class e.g. movement, origin for artists. The influence of the criteria selection is consequently

easily observable on this facet (but have also influence on the others). The top 10 result lists presented in the table 6.3 are all related to Claude Monet but constitute different topic-centered perspectives:

- The first query where no category of interest are specified retrieves artists that are strongly related to France and impressionism: 9 on 10 are French, 8 on 10 are impressionists.
- The second query where all the categories related to France were declared as uninteresting and the category *impressionist painters* declared of interest retrieves 9 non-French (and 1 French) impressionists painters. American artists, that are almost absent of the basic algorithm results, are well represented.
- The third query where all the categories related to France were declared interesting and the category *impressionist painters* declared uninteresting retrieve 10 French painters where 5 are not impressionists (realist, fauvist, romantic), 4 are impressionist but not only (fauvist, cubist, modern artist) and only 1 who is only declared as impressionist and post-impressionist.

Query	Claude Monet (1)	Claude Monet (2)	Claude Monet (3)
Criteria	None	Impressionist painters + Artists from Paris - People from Le Havre - Alumni of the École des Beaux-Arts - French painters -	Impressionist painters - Artists from Paris + People from Le Havre + Alumni of the École des Beaux-Arts + French painters +
Results			
1	Pierre-Auguste Renoir	Theodore Robinson	Pierre-Auguste Renoir
2	Alfred Sisley	Édouard Manet	Gustave Courbet
3	Édouard Manet	Alfred Sisley	Edgar Degas
4	Mary Cassatt	Władysław Podkowiński	Jacques-Louis David
5	Camille Pissarro	Leslie Hunter	Jean-Baptiste-Camille Corot
6	Edgar Degas	Theodore Earl Butler	Jean-François Millet
7	Charles Angrand	Lilla Cabot Perry	Paul Cézanne
8	Gustave Courbet	Frank Weston Benson	Marc Chagall
9	Berthe Morisot	Childe Hassam	Camille Pissarro
10	J.-Baptiste-Camille Corot	Edward Willis Redfield	Édouard Manet

Table 6.3: Results of three queries about Claude Monet using the criteria specification

As this algorithm variant does not modify fundamentally the functioning of the algorithm it was not the subject of extensive analysis. Its relevance is evaluated in chapter 8.

6.5.2 Controlled randomness variant

By nature the randomized version is divergent. As the objective is to disturb the original algorithm behavior in order to retrieve unexpected results we did not perform extensive analyses concerning its algorithmic behavior either. Its influence over the results and their perception by the users are evaluated in chapter 8. It is visible on the pseudo-code below that the randomized algorithm is not the same

if the chosen level of randomness is superior to 0.5 (lines 29 to 31) or lower (11 to 13).

```
   Data: o ∈ KB, maxPulse ∈ ℕ, limitImport ∈ ℕ, r ∈ [0,1]
   Result: activationHashMap(i,a); i ∈ KB; a ∈ ℝ
 1 begin
 2     activationHashMap.put(o, 1.0);
 3     getClassPropagationDomain(o);
 4     importNeighborhood(o) ;
 5     while i ≤ maxPulsedo
 6         foreach i in activationHashMap do
 7             if activationHashMap.get(i) > 0 then
 8                 for j in neighbor(i) do
 9                     sim = commontriple(j,o))
10                     act = activationHashMap.get(i) * (1 + sim) / degree(i)
11                     if r <= 0.5 then
12                         act = r * random() + (1-r) * act
13                     end
14                     tempMap.put(j, tempMap.get(j) + activation)
15                 end
16             end
17         end
18         activationHashMap = tempMap
19         foreach k in sortedByDecreasingValue(activationHashMap) do
20             if importSize>=limitImport then
21                 break() ;
22             else
23                 importNeighborhood(k)
24             end
25         end
26         i++
27     end
28     if r > 0.5 then
29         foreach i in activationHashMap do
30             activationHashMap.put(i, (r * random() + (1-r) * activationHashMap))
31         end
32     end
33 end
```

Algorithm 4: Monocentric semantic spreading activation pseudo-code, controlled randomness injection variant

The table 6.4 gives examples of query results with several level of randomness. The most important point to notice is that when the level of randomness is superior to 0.5[25] the successive neighborhoods imported are impacted and differ from the basis algorithm ones. In other words the propagation is highly susceptible to explore parts of the graph that are not reached with other query-configurations.

6.5.3 Data source selection

With the proposed software architecture it is easy to change the targeted SPARQL endpoint i.e. the linked dataset used to process the query. The differences between the results when using different DBpedia SPARQL endpoints are substantial. We executed the Claude Monet query on the 5 largest DBpedia chapters SPARQL endpoints. Regarding the *Artist* and *Museum* facets, both interesting because strongly associated to a country and culture, we observed that:

- Using the English DBpedia chapter, 4 artists and 5 museums in the tops 10 are from English-speaking countries (as officially recognized) i.e. the United Kingdom and the United States. Contrary to other languages the *Art Institute of Chicago* is ranked as the first museum (instead of the *Orsay Museum*).
- Using the French DBpedia chapter 9 artists are French in the top 10 artists and 9 museums are situated in French-speaking countries: 1 in Switzerland and 8 in France.
- The *Kunsthalle Bremen*, *Alte Nationalgalerie*, *Museum Folkwang*, *Wallraf-Richartz-Museum* and the *Fondation Corboud*, situated in Germany as well as the German artist *Max Liebermann* appears only in the German chapter results.
- The *Galleria nazionale d'arte moderna e contemporanea*, situated in Italy, appears only in the Italian chapter results.
- The *Botero* museum, situated in Columbia, appears only in the Spanish chapter results.

It is especially hard to evaluate the results relevance according to cultural criteria as it is profoundly subjective. Thus we decided to only evaluate quantitatively the cultural differences between the result lists obtained from different DBpedia local chapters. At the same time this experimentation was the occasion to evaluate the response-time in real conditions i.e. using third-party linked datasets. The following hypothesis was formulated:

- **Hypothesis 9**: Significant results variations exist when using different DBpedia chapter for the same query. In others words, the results reflect the knowledge variation present in the DBpedia chapters.
- **Hypothesis 10**: the response time is only few seconds when using third-party SPARQL endpoints.

[25] This value is discussed in chapter 8.

Randomness	**0**	**0.25**	**0.5**	**0.75**	**1**
Results					
1	Pierre-Auguste_Renoir	Mary_Cassatt	Édouard_Manet	Impressionism	Palette_(painting)
2	Edouard_Manet	Paul_Signac	Pierre-Auguste_Renoir	Anna_P._Baker	Canvas
3	Camille_Pissarro	Seine	Johan_Jongkind	Williamstown,_Massachusetts	Louis_Kahn
4	Edgar_Degas	1891_in_art	Alice_Hoschedé	Ukiyo-e	Marie_Bracquemond
5	Alfred_Sisley	Arsenic_poisoning	The_Song_of_the_Lark	Nicolas_Viel	List_of_architectural_design_competitions
6	Blanche_Hoschedé_Monet	Georges_Clemenceau	United_States_Academic_Decathlon	America'_Favorite_Architecture	Toi_gold_mine
7	Mary_Cassatt	Rolf_on_Art	Camden_Town_Group	Hetty_Burlingame_Beatty	Han_van_Meegeren
8	Gustave_Caillebotte	Pierre-Georges_Jeanniot	Musée_du_Louvre	Roman_roads	Divisionism
9	Frédéric_Bazille	Cecilia_Beaux	Blanche_Hoschedé_Monet	Edward_Dugmore	List_of_most_expensive_paintings
10	Charles_Angrand	Jean-Baptiste_Faure	Great_Chicago_Fire	Assassin's_Creed:_Brotherhood	House_of_Representatives_of_Japan
Neighborhood loaded					
	Gustave_Caillebotte Blanche_Hoschedé_Monet Alfred_Sisley Pierre-Auguste_Renoir Édouard_Manet Frédéric_Bazille Charles_Angrand Alice_Hoschedé Johan_Jongkind Mary_Cassatt Jacques-François_Ochard Adolphe-Félix_Cals Paul_Durand-Ruel Camille_Doncieux Edgar_Degas Eugène_Boudin Suzanne_Hoschedé Camille_Pissarro			Le_Bassin_Aux_Nymphéas Art_Institute_of_Chicago Normandy	Kunstmuseum_Winterthur Camille_Doncieux Vétheuil Chichu_Art_Museum Alice_Hoschedé Epte Johan_Jongkind Palace_of_Westminster Eugène_Boudin Houses_of_Parliament_series_(Monet) Naoshima,_Kagawa Museum_Boijmans_Van_Beuningen Marc-Charles-Gabriel_Gleyre Le_Bassin_Aux_Nymphéas Alfred_Sisley Women_in_the_Garden Snow_at_Argenteuil Blanche_Hoschedé_Monet Barnes_Foundation Kimbell_Art_Museum Paul_Durand-Ruel National_Gallery_of_Scotland

Table 6.4: Results of 5 queries using different having different levels of randomness injected

In order to verify this hypothesis we first filtered the whole list of distinct queries entered in Discovery Hub (2302)[26] to keep only the entities that were described in the 5 biggest DBpedia chapters: the English, French, German, Italian and Spanish ones (all of them have over 1 million resources). The amount of query-entities that were described in all this 5 chapters was 739 (32%)[27]. Then we processed the query with the 5 SPARQL endpoints (the localized DBpedia versions) and compared the French, German, Italian and Spanish chapter results with the English chapter ones. We chose to compare them with the English chapter ones because the vast majority of existing applications use it and only it today. The results are shown on Figure 6.17. It is interesting to notice that the top 100 shared results are relatively low and that a consequent proportion of these results do not exist in the English DBpedia chapter. Thus the hypothesis 9 is verified. The average execution time on each chapter was few seconds (maximum 5 seconds for the English chapter and minimum 3 seconds on average for the Spanish one). It shows that the framework has interesting performances using public online SPARQL endpoints, the hypothesis 10 is verified.

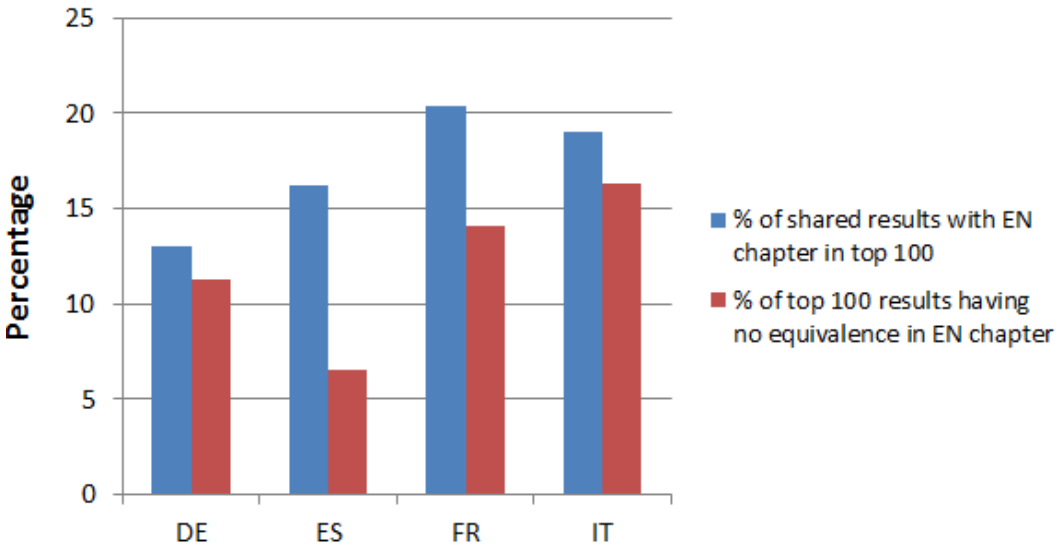

Figure 6.17: Percentage of shared results with the top 100 English chapter results and percentage of top 100 results that are specific to the chapter

6.6 Other datasets and calibration generalization

We will now discuss the applicability of the algorithm outside the DBpedia context. Indeed, together the DBpedia chapters constitute only a small portion of the entire Linked Open Data cloud. Moreover they have a very interesting knowledge

[26] http://discoveryhub.co/querylog-DH.txt
[27] http://discoveryhub.co/querylog-DH-multilingual.txt

coverage due to their encyclopedic nature but for some explorations it would be more interesting to target more specialized knowledge base e.g. Drugbank in a pharmaceutical research context.

6.6.1 Random graphs

In order to understand the behavior of the algorithm outside DBpedia we first performed an analysis on a variety of graphs generated on-purpose and having diverse properties. For this we used the Graphstream[28] library that supports the generation of random graphs according to various properties. As Graphstream does not support natively the generation of RDF graphs we used the following process:

- Generation of the desired graph with the GraphStream library in GEXF format[29].
- Transformation in ntriple format thanks to a JAVA script we coded.
- Transformation in RDF/XML syntax thanks to the RDF2RDF library[30].

We verified that the graph transformation from GEXF to RDF/XML format was perfectly executed. For this we checked that we found the same number of nodes, average degree, edge density and edge variance on the GEXF graph using the Graphstream library and on the RDF/XML conversion using Kgram functions. The comparisons showed that the conversion process was perfectly preserving the test graphs among the successive steps.

The Graphstream library proposes several generators that output specific graph structures. They are detailed on the online documentation[31]. During a first round of analysis we used several generators: the Random, Barabasi, Dorogovtsev and Mendes, Small world and Random Euclidian ones. It rapidly appears that the generators did not have a strong influence over the algorithm behavior, but that some metrics of the graph did. Thus we decided to use only the random generator during the second round of analysis.

3719 random graph were generated (one night of processing), and one monocentric spreading was executed for each of them by stimulating a randomly selected node. All the graphs were composed of 1000 nodes and a varying amount of arcs. During our preliminary analyses we noticed an unclear dependance between the average graph degree and the convergence of the algorithm. In order to set the degree range of our analysis we referred to the Koblenz network collection (KONECT). KONECT is *"a project to collect large network datasets of all types in order to perform research in network science and related fields"*. This collection gives indications about the characteristics of real-world networks. In KONECT more than 70%

[28] http://graphstream-project.org/
[29] http://gexf.net/format/
[30] http://www.l3s.de/m̃inack/rdf2rdf/
[31] http://graphstream-project.org/doc/Generators/

of the networks have a degree comprised between 2 and 30[32] [105]. Thus we chose to make the average graph degree vary between 2 and 30.

It has to be noticed that such graphs are untyped and can not simulate the influence of the semantics over the algorithm behavior. We can considered that the $CPD(o)$ filtering operation was already applied but in any case the $commontriple()$ similarity measure is not simulated. The figure 6.18 hereafter presents several metrics distributions of the generated graphs. It is observable that the distribution of the average degree, the density, variance and the clustering coefficient of the graphs are linear. Otherwise the figure 6.19 shows that the distribution of the graphs diameter is not linear. The average value for all these metrics are given below:

- Average degree: 15.72.
- Density: 0.015.
- Variance: 15.33.
- Clustering coefficient: 0.015.
- Diameter: 5.9.

As mentioned previously we noticed a dependence between the average degree and the convergence of the algorithm. Nevertheless it was unclear and we rapidly searched for another metric to explain the variations we observed in the number of iterations needed by the algorithm to converge. Due to its dependence with the degree and its logical impact on the behavior of the traversal algorithms the influence of the diameter was more particularly investigated. The figure 6.20 shows the relation between the average degree and the diameter of the generated graphs. Unfortunately the diameter is not a parameter that is controllable with the GraphStream generators. Consequently we have diverse amounts of graphs for each diameter (between 2 and 33), see table 6.5.

Diameter	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15+**
Occurrence	264	1740	638	290	164	128	89	64	76	21	27	49	169
Percentage	9.7	63.99	23.46	10.66	6.03	4.7	3.27	2.35	2.79	0.77	0.99	1.8	6.21

Table 6.5: Proportions of graphs per diameter ranges

Then we performed analyses similar to the one we did for the monocentric query. For each graph we randomly stimulated a node and executed a spreading activation algorithm. Then we observed its convergence by computing the amount of top 100 shared results from an iteration to another. We also computed the τ_b rank correlation coefficient of these shared results. For sake of clarity we grouped the diameters by 2 from 3 to 10, then we grouped the diameter between 10 and 15 together, and finally we grouped the diameters superior to 15 together. The results are presented on the figure 6.21 (shared results) and 6.22 (Kendall-Tau). We

[32]The author thanks Jérôme Kunegis for his help and information

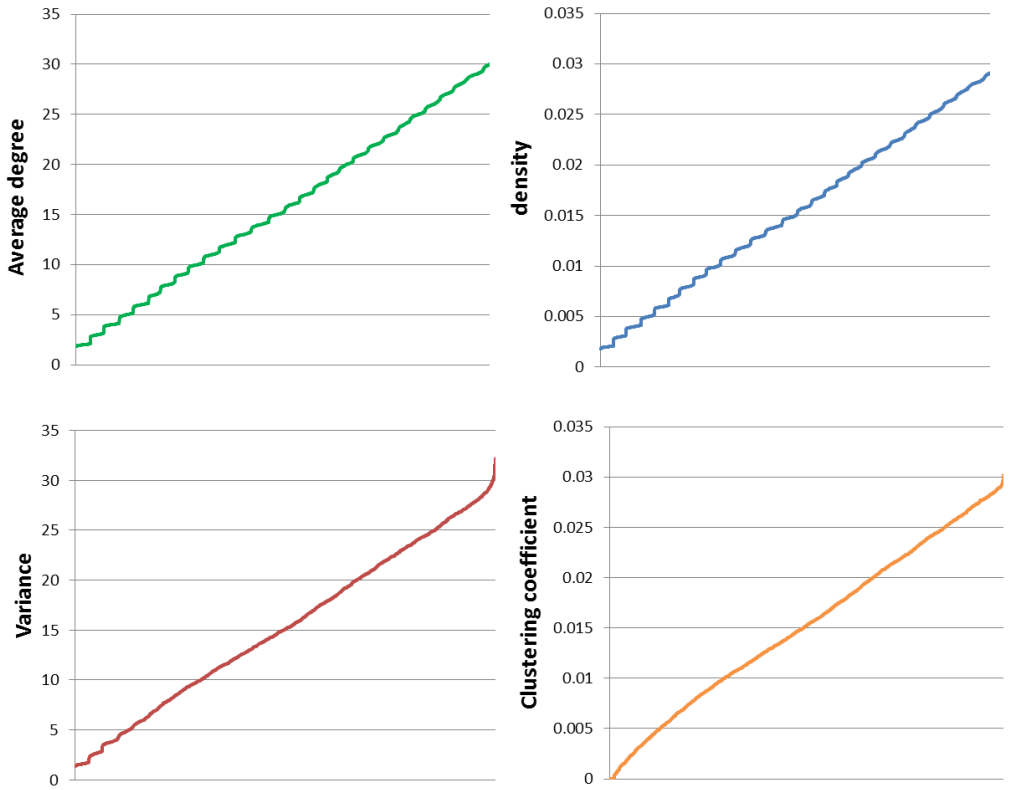

Figure 6.18: Principal metrics of the random graphs

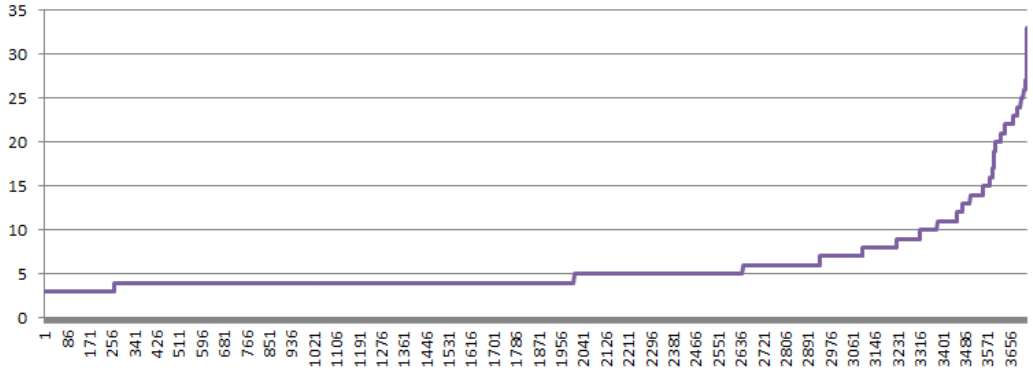

Figure 6.19: Diameters distribution of the random graphs

observe that the algorithm convergence is dependent on the diameter of the processed graph. The fast convergence in the case of DBpedia is due to fact that we partially import a graph that already has a low effective diameter of 6.27 (see appendix A on page 233). The effective diameter is defined *as the minimum number of hops in which 90% of all connected pairs of nodes can reach each other.* [145]. Considering spreading activation the graph effective diameter is more informative than

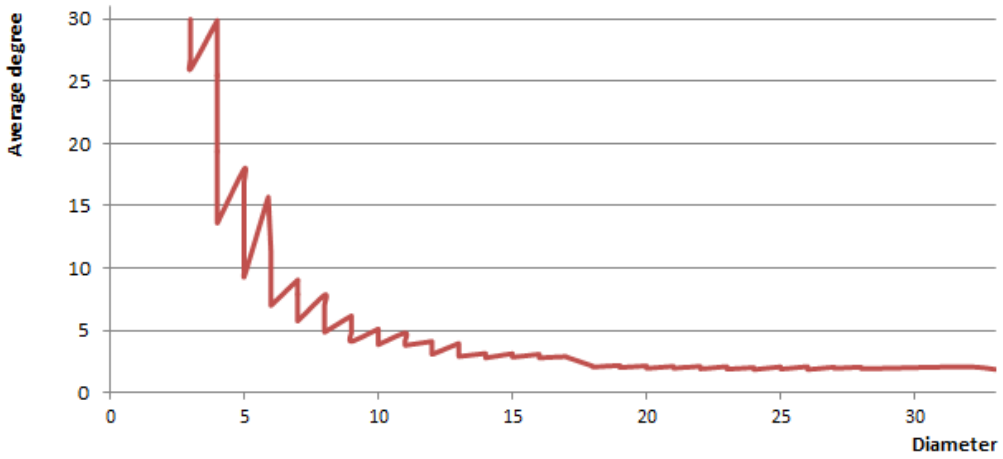

Figure 6.20: Relation between the graphs average degrees and diameters

the diameter that can be significantly high due to pathologic structures. The τ_b fall that can be observed for the high diameters is due to the density of the graph. Indeed the τ_b rank correlation coefficient is computed on the shared results between the iteration $n-1$ and n. For the graph having a high diameter only few nodes are activated during the first iteration, there is few ranks movements so τ_b is high. However the number of nodes activated is more important at the next iteration, there are more movements in term of ranks thus τ_b decreases. Then we observe that the values converge as τ_b constantly increases.

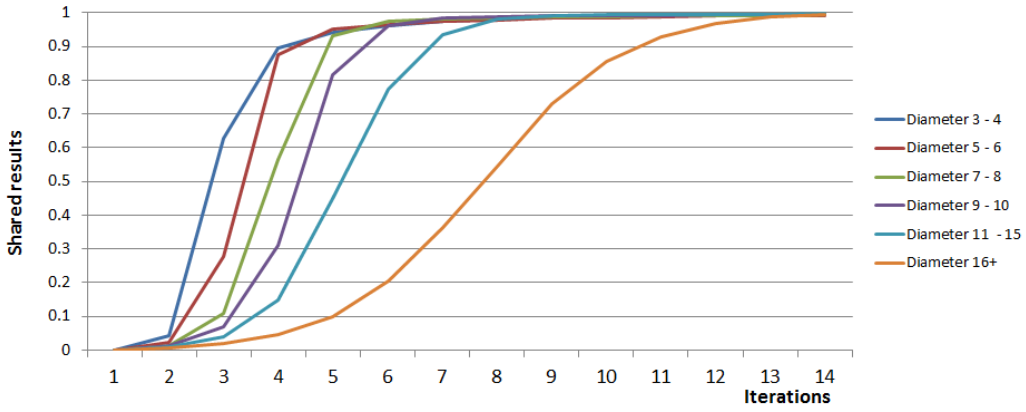

Figure 6.21: Amount of top 100 shared results from one iteration to another ($n-1$, n) according to the graph diameter

The KONECT networks collection[33] has an average effective diameters comprised between 0.89 and 698.9 with an average of 13.55 but only 5.65 if we do not consider the 3 longest diameter (that are over 500). A histogram of the effective

[33]http://konect.uni-koblenz.de/networks/

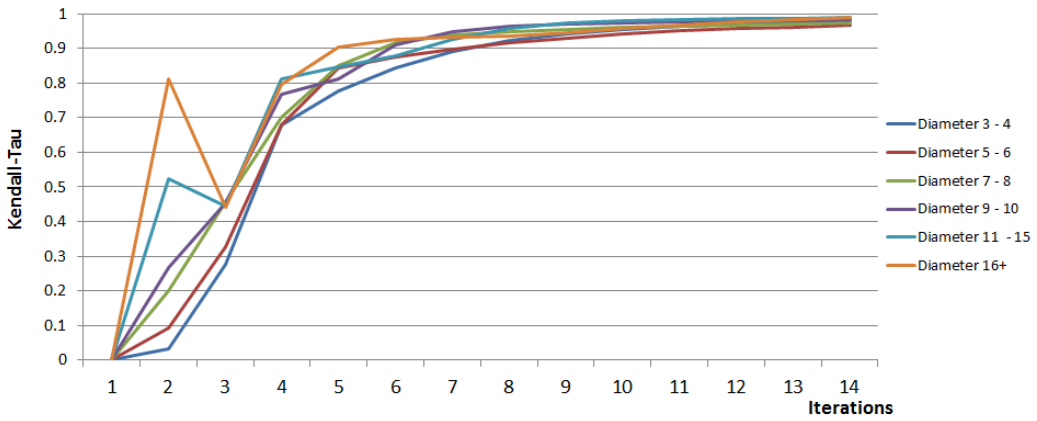

Figure 6.22: τ_b of the 100 shared result from one iteration to another $(n-1,\ n)$ according to the graph diameter

diameter is presented on the figure 6.23. Generally the networks have an effective diameter that is close to the DBpedia one. Thus the algorithm is potentially applicable on a wide range of real world graphs.

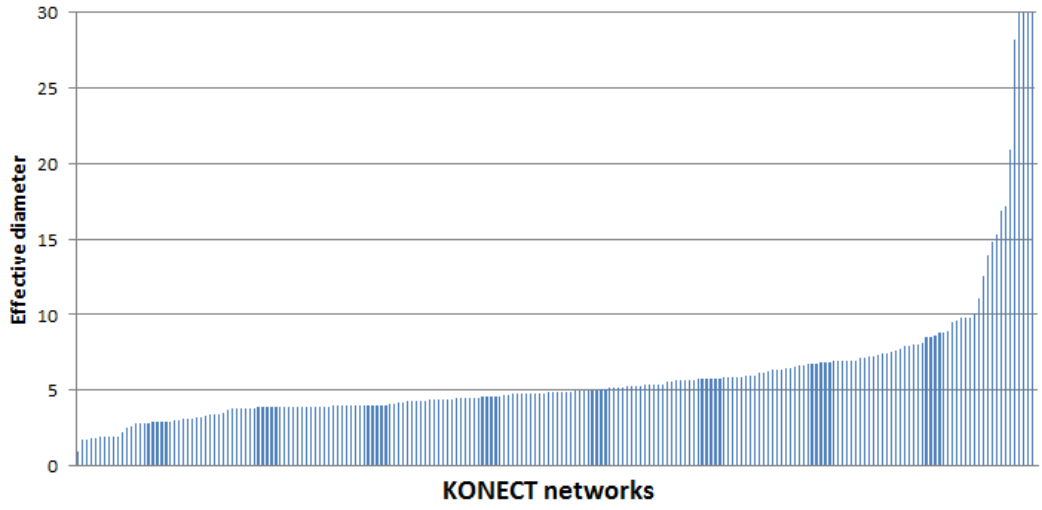

Figure 6.23: Histogram of the effective diameters of the 219 networks of the Koblenz networks collection

6.6.2 Digg dataset

Finally we performed an analysis on another real-world dataset a Digg[34] RDF dataset[35]. We chose this dataset in particular because social networks are especially interesting for the application of the algorithm. They are often large, het-

[34]http://digg.com/

[35]http://konect.uni-koblenz.de/downloads/rdf/munmun$_{d}igg_{r}eply.n3.bz2$

erogeneous and evolve in real-time. More precisely the RDF dataset is the "*reply network of the social news website Digg. Each node in the network is a user of the website, and each directed edge denotes that a user replied to another user*"[36]. It has a diameter of 12 but a 90 percent effective diameter (also known as effective diameter) of 5.40 [37]. It is constituted of 30.398 nodes (users) and 87.627 edges (replies).

We run 12.000 monocentric queries on this dataset (one night of processing) starting from a randomly selected node each time. We used the incremental import strategy. According to the metrics of the Koblenz network collection the Digg dataset 90 percent effective diameter is very close to the DBpedia one. Thus we reused the import limit of 6000. The figures 6.24 and 6.25 show that the algorithm has a behavior similar to the behavior it has over DBpedia. We observe the τ_b fall. It is not a surprise as the dataset has a lot density[38] (an average degree of 5.7653 edges / vertex). The algorithm appears to be applicable on the Digg social network structure with the previously identified parameters.

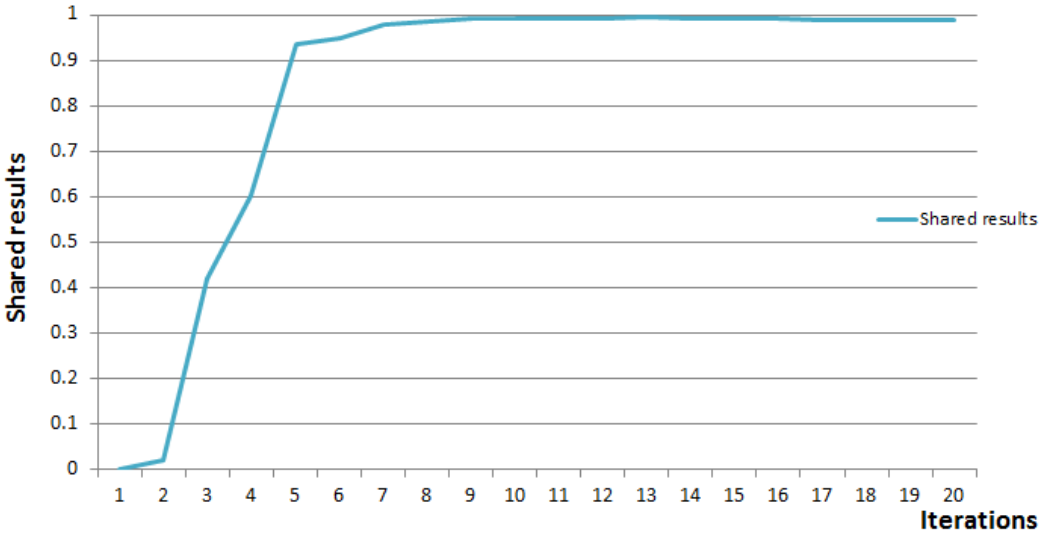

Figure 6.24: Amount of top 100 shared results from one iteration to another ($n-1$, n), Digg analysis-case

6.7 Conclusion

In this chapter we detailed and motivated the software architecture we chose in order to reach an unprecedented level of flexibility in algorithmic linked data based exploratory search. We proposed to compute the results at query-time from distant data imported from online SPARQL endpoints. This software architecture brings multiple research challenges. Indeed, due to data providers service limits, transfer

[36] http://konect.uni-koblenz.de/networks/munmun$_{d}igg_{r}eply$
[37] http://konect.uni-koblenz.de/networks/munmun$_{d}igg_{r}eply$
[38] http://konect.uni-koblenz.de/networks/munmun$_{d}igg_{r}eply$

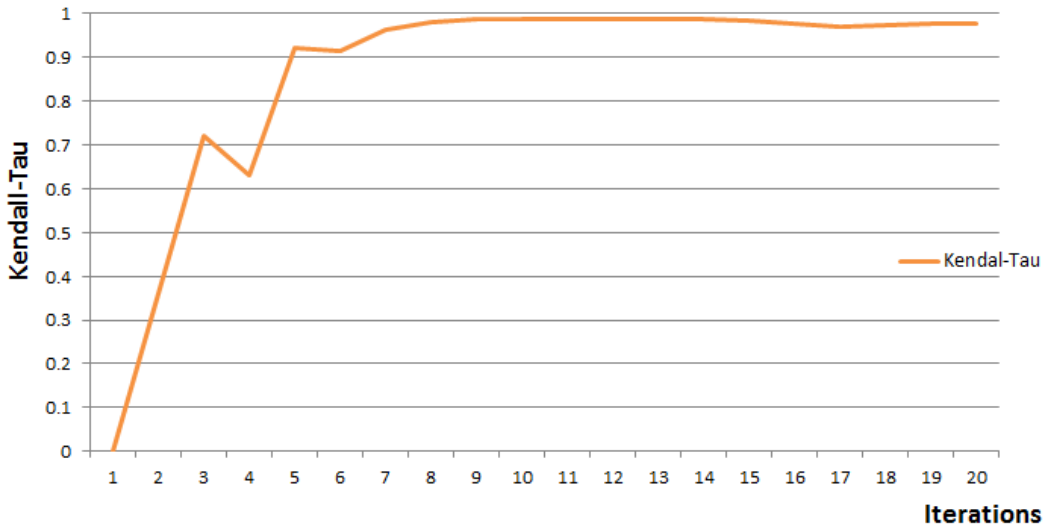

Figure 6.25: Kendall-Tau of the 100 shared result from one iteration to another $(n-1, n)$ according to the graph diameter, Digg analysis-case

and computational costs the algorithm should be run on a small amount of data. A triangular trade-off appears between the size of the import, the computational cost and the quality of results. We studied the behavior of the algorithm for both mono-centric and poly-centric queries. Thanks to extensive analyses we identified the best settings for our implementation over DBpedia. We also produced a variety of algorithm visualizations for observation and communication purposes, see appendix C. Some of these visualizations are also available online in the form of a video[39].

The execution of a set of queries on the English, French, German, Italian and Spanish SPARQL endpoints showed considerable differences between the result lists. It is also noticeable that an important part of the results does not exist in the English DBpedia chapter. Thus it is important to pursue the efforts of the DBpedia internationalization both in quantity and quality to avoid an over-use or misuse of the DBpedia English chapter. At the end of the chapter we discussed the applicability of our algorithm and approach on other graphs. To this end we studied the algorithm behavior over randomly generated graphs and over a Digg dataset. According to this analysis that the algorithm convergence depends mainly on the graph diameter. Thus it is probably possible to apply the algorithm on other LOD graph as their diameters are close to the DBpedia one.

Now that we are able to identify a result set informative about a topic or a topics combination we will present the interaction model and the interface we propose for their exploration in the following chapter.

[39]https://www.youtube.com/watch?v=_Lc-TDMpxnI

CHAPTER 7

Designing user interaction for linked data based exploratory search

Contents

7.1 Introduction

In this chapter we present the interface and the interaction model of the web application based on our framework: Discovery Hub[1]. The exploratory search engines are sometimes referred to as to *Human-Computer Information Retrieval systems*

[1]The web interface was designed with and entirely developed by the former masters' degree student Damien Legrand (http://damienlegrand.com/) during 2 internships. The author thanks him again for his major contribution to this thesis

(HCIR) by the scientific community. This denomination emphasizes the importance of the interactions for successful explorations. The interfaces have to be optimized to favor the users' engagement and to support them continuously and intensively on a cognitive point of view.

As exploratory search is executed through lengthy successions of heterogeneous tasks the systems interfaces are often rich in terms of functionalities. Such functionalities are gathered in an interaction model. The result forms a complex alchemy that is subject to the tension between the interactions intuitiveness and precision. With the Discovery Hub application our aim is to favor the intuitiveness at the maximum. We want to develop a system that can be used by a wide range of users because exploratory search systems can be useful to everybody e.g. for educational learning, professional decision-making or leisure explorations.

The interface aims to help the searchers at every step of their explorations. It offers functionalities that both support exploratory tactics (algorithmic-orienteering, faceted mechanisms) as well as focused search ones (rich resources pages, explanatory features). Discovery Hub is designed to support diverse users and their corresponding search behaviors. For instance, they have the choice in several browsing modes and can access three very different result explanatory functionalities. Each explanation feature offers a different perspective on the data.

We leveraged the data semantics in the application design. The semantics are very useful at the interface level where they can be used to structure the interactions and the understanding. In other words the data semantics is an important factor of sense-making. Moreover, when using DBpedia as the primary source of knowledge, its correspondence with Wikipedia can be leveraged to raise the users' understanding.

In this chapter we will review (7a) the Discovery Hub interface, with a focus on each of its important components, (7b) we discuss the Discovery Hub interface by comparing it to the first version, we evaluate its compliance regarding the existing guidelines as well as its correspondence with the systems desired effects we identified in chapter 2, (7c) we present the context of the interface development. The Discovery Hub application was demonstrated at ESWC2013 [127] and ISWC2014 [124] conferences. It won the best demonstration award of ESWC2013[2].

7.2 Interaction design

It appeared after its conception that the interaction model can be analyzed under the often-cited "*information seeking mantra*" proposed by Ben Shneiderman [167]. It has to be noticed that we were unaware of the existence of this set of guidelines when we designed and developed the Discovery Hub interface. It appeared later that the interaction model we developed particularly fits it. It is consequently used here-after for presentation purposes. The information seeking mantra is a high level guideline for designing information visualization applications. His author,

[2]http://2013.eswc-conferences.org/news/brief-summary-exciting-10th-eswc2013

Ben Shneiderman, presented it as an inspirational asset rather than a prescriptive one. The following description is taken from [167]:

- **Overview**: gain an overview of the entire collection.
- **Zoom**: zoom in on items of interest.
- **Filter**: filter out uninteresting items.
- **Details-on-demand**: select an item or group and get details when needed.
- **Relate**: view relationships among items.
- **History**: keep a history of actions to support undo, replay and progressive refinement.
- **Extract**: allow extraction of sub-collections and of the query parameters.

This simple set of guidelines has a certain success outside the information visualization community. At the time of writing 89 contributions focused on exploratory search cite it, according to Google Scholar[3]. To the best of our knowledge the applicability of the information seeking mantra to exploratory search systems was not extensively discussed. The correspondence between its components and the Discovery Hub functionalities is listed here-after:

- **Overview**: results list.
- **Zoom and filter**: results list faceted mechanisms.
- **Details-on-demand**: result pages.
- **Relate**: explanatory features.
- **History, extract**: user profile.

7.2.1 Homepage

Today, the exploratory search systems can appear obscure for some users that are accustomed to the widespread lookup-optimized search engines. Consequently there is a need to explain the objective of such tools and what can be the benefits for the users. Consequently we built a short tutorial that appears during the first connection to the website, see figure 7.1. The tutorial explains what is exploratory search, how to start an exploration, how to get explanations about the results and also presents on the social mechanisms of the application. The same information are shown persistently on the homepage with a more descriptive text in 4 rubrics: *exploratory search*, *get recommendations on your interests*, *understand a recommendation*, *share your findings with your friends*.

The large search bar with the text *start your exploration here* is the homepage *call-to-action*[4]. On the top-left the *news* and *random* pages aim to incite the users to test the application. The *News* page displays a subset of news that are semantically annotated using the Reador API[5]. The *Random* page shows a subset of queries

[3]http://scholar.google.com
[4]http://thelandingpagecourse.com/call-to-action-design-cta-buttons/
[5]https://app.reador.net/

previously entered by the Discovery Hub users in the form of a pictures mosaic.

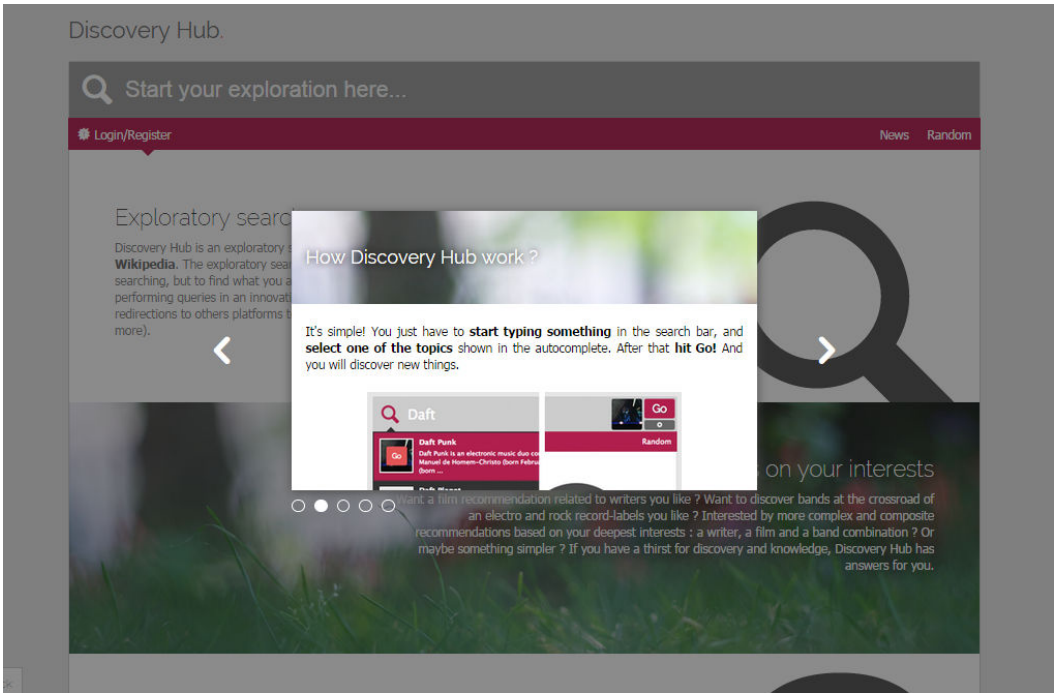

Figure 7.1: The homepage, a short tutorial is shown during the first visit

7.2.2 Querying

The query-bar emphasizes the exploration purpose of the search in Discovery Hub i.e. *start your exploration here*. It uses the DBpedia lookup API[6] which offers rapid resources selection using typing auto-complete, see figure 7.2. For each resource suggested their name, picture, abstract and class are shown for a more precise selection. It is also possible to filter the resources suggestions by their classes at the bottom. The users can directly launch the query by clicking the *go* button. They can also click elsewhere on the resource zone to build composite queries and reach the advanced querying functionalities, see figure 7.3.

7.2.3 Browsing

Once the query is processed an informative result set related to the topic(s) of interest is retrieved and available for exploration. The Discovery Hub search action clearly supports an orienteering behavior. Orienteering is the practice of entering vague queries retrieving a lot of potential relevant results. It is followed by a naive navigation of the results space that is progressively replaced by more structured and analytical tasks as the information exposure raises the users' understanding.

[6]http://wiki.dbpedia.org/Lookup

Figure 7.2: The search bar with rich resources presentation and filtering option

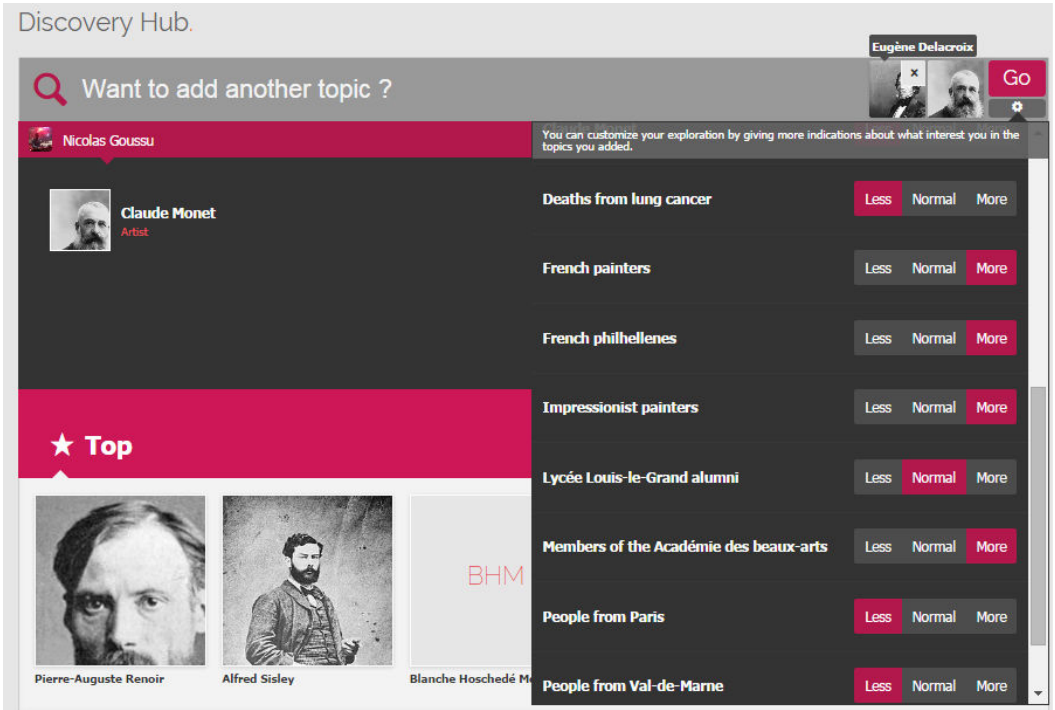

Figure 7.3: Search bar advanced querying functionalities

Depending on the users and the information needs such orienteering behavior can be used to retrieve entrance points (results pop-up) for structured sequences of interactions (focused search) or being used repetitively on-purpose to favor dis-

coveries, see the appendix I on page 253.

7.2.3.1 Results list

When the query is processed the results list is retrieved in a page optimized for its exploration, see figure 7.4. The results list in semantic search system is a critical component. Indeed the display of the results reflects the developers' choices of what data combination constitutes the answer to the users' information need. In our case we made interaction and design choices that aimed to support exploratory tactics in a first time. The Discovery Hub result page corresponds to the **overview** component of the information seeking mantra.

We wanted the results list to be very visual for both esthetic and overview concerns. Thus the results are presented in a structured mosaic using their images. This *paints a picture* of the related topics that serves for the visual identification of relations, patterns and gaps. By relying on their visual ability the users can quickly increase their understanding of the domain and shape/refine their information need. In order to highlight the diversity of the results they are presented in the form of horizontal blocs. Each bloc corresponds to a class from the class-propagation domain. The class name is explicitly displayed on the interface. Each of the blocs contains 40 results at the maximum. At the top appears a *top* results list which is composed of the 40 most activated resources independently from their classes. In the actual configuration 12 blocs are shown: the top one and the eleven most prevalent classes in the class-propagation domain including the *miscellaneous* one. We chose to display 12 blocs after several HTML mock-ups; the objective was to display a high amount of results without showing an excessively long results list. The miscellaneous bloc contains the resources having a less prevalent type in the CPD as well as the untyped resources.

Such CPD classes are leveraged to offer faceted browsing in the results list. Note that they are permanently displayed at the top for rapid scroll by clicking, see figure 7.4. They are alphabetically ordered. We tried to rank them by order of prevalence in the neighborhood (using $NT(o)$, see chapter 5) but it resulted in hardly interpretable results. Indeed in the case of Claude Monet using such prevalence order would lead to *Artist* and *Museum* at the 2 first positions which is coherent. But it would also lead to *Person* at third position followed by *Book* and then *River*. This order is confusing and it resulted in loss of time for the class position identification on the interface.

Discovery Hub has a dynamic information architecture. The facets structuring the interface are based on the class-propagation domain and are consequently query dependent. This aspect is especially valuable for cross-domain exploratory search systems where there is a need to expose different elements of information regarding the query nature. A second level of facets is available for each *class-facet*, in order to support faceted-filtering[7], see figure 7.5. This corresponds to the

[7] at the time of writing this functionality was not implemented yet on the V2, it will be available on-demand to avoid information overload

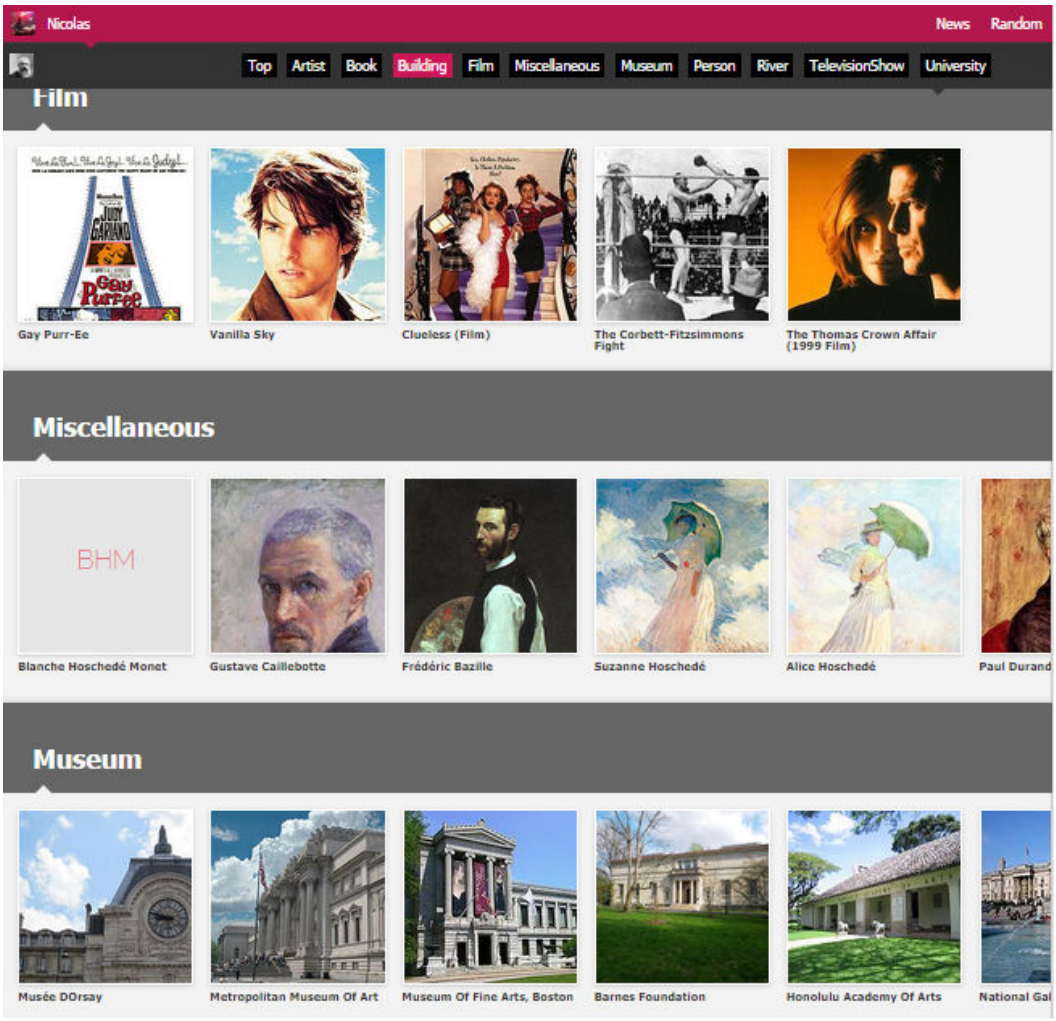

Figure 7.4: Result list

Figure 7.5: Filters associated to the Claude Monet *Film* facet

zoom and filter components of the mantra, which *"reduce the complexity of the data representation by removing extraneous information from view and allow for further data organization"* [35]. The users' cognition is assisted by selectively hiding or unveiling specific results of interest. Unexpected relations among the results might also appear during this process. This second level of facets corresponds to the most prevalent DBpedia categories associated to the results of a class/bloc e.g. the films. In order to ease the selection of the filters we introduced a color code: rare DBpedia categories are displayed brighter e.g. *museums in popular culture*. The narrow categories are assumed to be more informative than the wide ones e.g. *2001 films*. Such filtering-facets are identified thanks to the SPARQL query 7.1.

```
1  select  ?categories where {
2   service <sparqlendpointurl>
3   {
4    select ?categories (count(?x) as ?count) where {
5     ?x <http://purl.org/dc/terms/subject> ?categories
6     filter( ?x = result1Facet1  || ?x=
7      result2Facet1 || ?x = result3Facet1 ... )
8    } order by desc (?count)
9   }
10  }
```

Listing 7.1: SPARQL query to identify the categories related to a class-facet

7.2.3.2 Results page

When the users click a result they turn into a focused search task that requires more precise information. An important design choice in the actual Discovery Hub version was to show the results page as pop-ups appearing on top of the results list, see figure 7.6. The objective was to avoid a disruption between the results list and the consultation of a specific result. Using a pop-up keeps intact the informational context in which the result is situated: the horizontal and vertical scrolling positions are not lost for instance. This functionality corresponds to the information seeking mantra **details-on-demand** component which *"provide additional information on a point-by-point basis, without requiring a change of view"* [35].

The result pop-up shows all the important information about the resources such as their title, picture and full abstract. On the left it displays structured information constituted of DBpedia properties and their associated objects e.g. *movement impressionism*. Clicking them it is possible to browse the DBpedia resources in a classic one resource-at-a-time semantic browsing approach. In this case a breadcrumb appears on the top of the pop-up, allowing to track the sequences of browsing and to come back to a previous step. It is also possible to switch back to an orienteering tactic by clicking the *run an exploration* button in order to trigger the algorithm again. This button is an example of the bridges we built to easily switch from a focused search strategy to a more exploratory one.

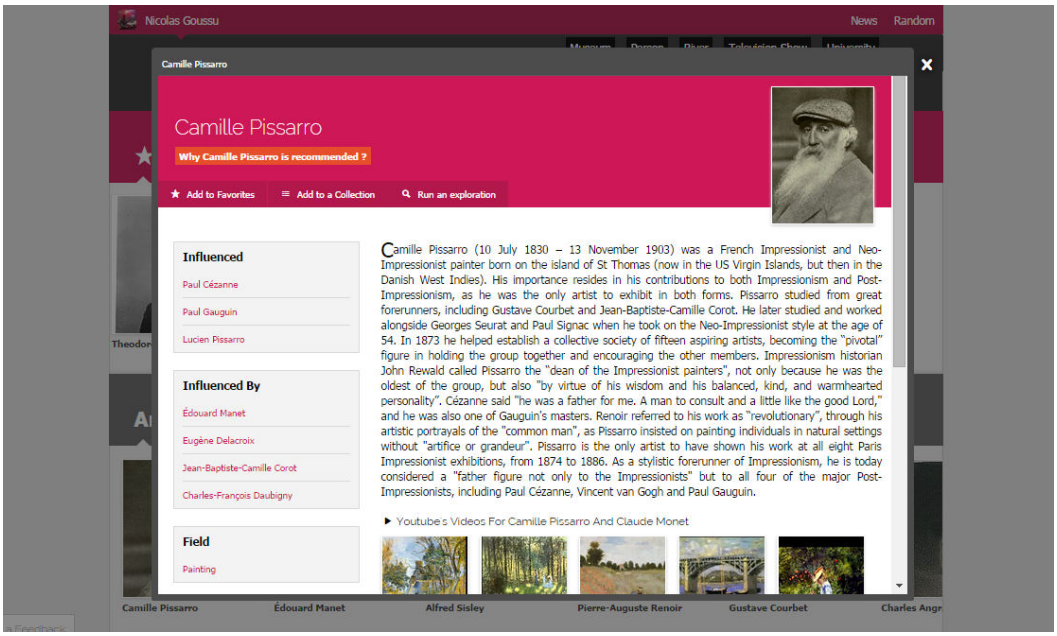

Figure 7.6: The result pop-up

The application was called Discovery Hub because the objective is to redirect the users to relevant third-party platforms once they have made discoveries. The exploratory search systems must *" reward users for effort expended"* and must *"be integrated into the information ecology rather than acting as discrete stand-alone services"* [194]. To this end we associated services to DBpedia classes. For instance the class *Museum* can be associated to the services TripAdvisor, Google Art Project[8], Lonely-Planet[9], etc[10]. Several services such as Youtube or Twitter are more generic and can be proposed for all the resources. In the current Discovery Hub implementation the services are integrated at the maximum in the interface instead of redirecting the users using query-urls including the resource label[11]. For instance the Sound-cloud songs are directly played in the Discovery Hub interface using the music streaming application API.

An important point is that the result pop-up supports two account-related interactions for saving and organizing the resources discovered. The first one is the *add to favorites* action which saves the resource in the user profile for further re-use. The second is a bit more sophisticated and allows the users to add the resources into collections. The collections are set of resources that the users gather about a high-level interest along their successive search sessions. These functionalities are detailed in the following *user profile* subsection. Finally the button *"why this*

[8]https://www.google.com/culturalinstitute/project/art-project
[9]http://www.lonelyplanet.fr/
[10]See redirections example on the screencast: https://www.youtube.com/watch?v=iM-WKcdbuEA
[11]e.g. http://www.tripadvisor.fr/Search?q=mus%C3%A9e%20du%20louvre

resource is recommended?" allows to ask for explanations about the results. The explanations functionalities are detailed in the following subsection.

7.2.4 Explanatory features

The explanations are critical sense-making factors as they help the continuous understanding of the resources connections. The results retrieved during an exploratory search session should be explained to the users for several reasons. First the searchers often lack knowledge about the informational domain they explore. They consequently need the results to be explained to rise their understanding and to shape new useful mental frames. Second some systems, including Discovery Hub, have specific functionalities that aim at retrieving unexpected results. Such results should be particularly well-explained to the users as they can be non-obvious. Retrieving unexpected results can favor discoveries but comes with the risk of degrading the users' perception of the results. Third, such explanatory features minimize the loss of confidence when the algorithms retrieve irrelevant results. Indeed if the users understand the reason why an irrelevant result appears they will be less disturbed. Fourth the explanations can bring extra-knowledge by presenting the resources and their relations through an angle that is not visible in the other application functions such as the result lists or pages. They can even be a source of inspiration that modifies the users' information need.

In Discovery Hub when the users are interested or intrigued by a result, they can ask for three different explanations thanks to three distinct features. This diversity in term of explanation has several reasons. Proposing several explanations can support the understanding of diverse users that can have preferences regarding their mental frames. Moreover the explanations complete each other as they all have their strengths and weaknesses. Last but not least some explanation methods we use only produce results under certain conditions. For instance, the Wikipedia-based one requires the resources to be direct neighbors in DBpedia.

The three features are detailed below. When the users click on the result pop-up *"why this resource is recommended?"* button all the explanations are computed. The ones that produce a result are shown on top of the result abstract. The explanations correspond to the **relate** component of the information seeking mantra. They are detailed below:

7.2.4.1 Common-triples explanation

The first explanation shows the set of property-values combination the query-resource(s) and the result-resource have in common, see figure 7.7. They are shown in a natural language text using a simple template: *"the [resource X] and the [resource Y] are both [shared categories,] and have the same [properties label: objects,]"*. This presentation in a natural language form enhances the users' understanding. Indeed in [147] the template explanation *"[resource X] and the [resource Y] share the same value for the [property label]"* was considered as *"useful"* by 9 out of 10 participants

but also *"too geeky"* by 6 of them. The common categories are listed in the first place because they have an important impact on the results ranking in the current implementation, due to the $commontriple(i,o)function$.

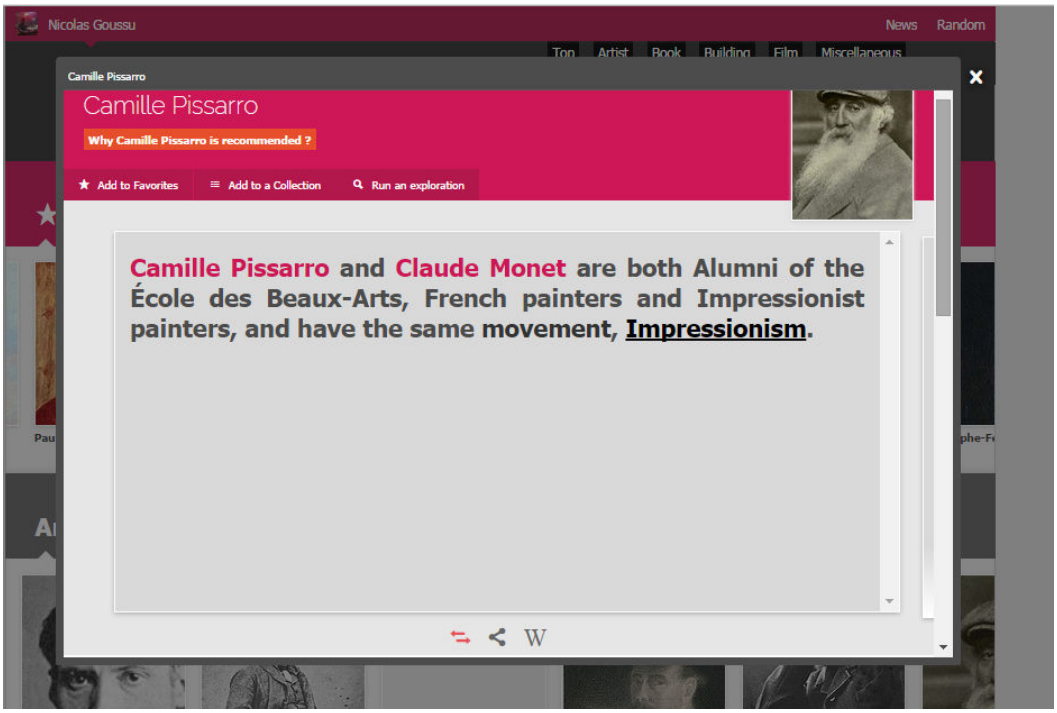

Figure 7.7: The common triples explanation

7.2.4.2 Wikipedia cross-references explanation

The second explanation leverages the link between Wikipedia and DBpedia. It identifies and retrieves the resources cross-references in their Wikipedia pages, see figure 7.8. The corresponding paragraphs are extracted and the resources are shown in bold and color. This explanation is very powerful to finely understand the relation between the resources but it only produces a result if the resources are direct neighbors in DBpedia. However the algorithm often retrieves *distant* results as discussed in chapter 6. This was the main motivation to propose a third explanatory functionality: the graph-based one.

7.2.4.3 Graph explanation

The third explanation shows the relations between the result and the query-resource(s) in a graph form, see figure 7.9. This view on data helps the exploratory searcher to understand some relations and to increase its knowledge by discovering visual patterns between the resources of interest. As mentioned by [40]: *"the power and value of visualization is seen in its ability to foster insight into and improve*

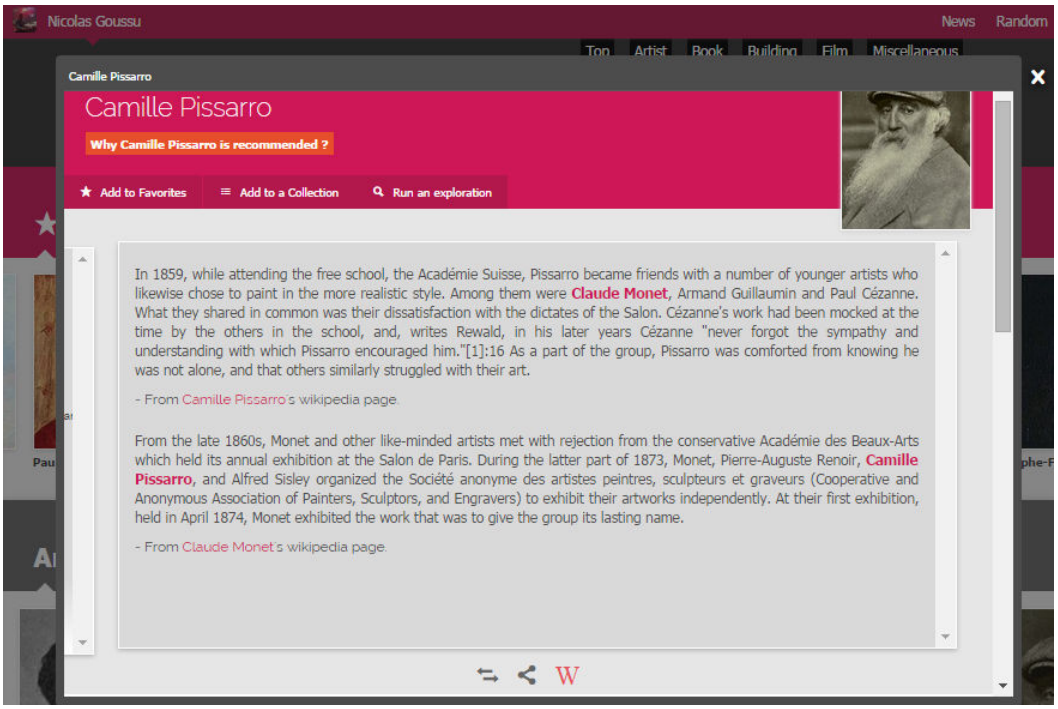

Figure 7.8: The Wikipedia cross-references explanation

understanding of data, therefore enabling intuitive, effective knowledge discovery and analytical activity". It is particularly adapted to polycentric queries as it allows to observe the resources being at the cross-road of all the selected resources (query ones and results). It can also inspire the users and encourage them to explore new topics as it unveils numerous elements of context. When the users go over a node its abstract appears on bottom of the graph. Its neighbors are circled with color in order to observe the relations existing in the graph. The position of the nodes can be changed using drag-and-drop. The Data Driven Document (D3) JavaScript library[12] is used to display the graph.

This graph is built on demand thanks to an algorithm. Its values have been experimentally set, based on performances and display clarity concerns. First an indirect path query (distance 2) between the resources is sent to the SPARQL endpoint, the same as the one used for composite querying described in chapter 6, see page 153. If this query is unable to retrieve 20 nodes we use the same protocol as for polycentric queries: we send directed path in both direction with a distance of 3 and 4 if necessary. Once 20 resources are retrieved they are sorted first by increasing order of distance, second by increasing order of degrees. Finally the CONSTRUCT query 7.2 is executed to build a graph that includes from the resources identified at the crossroad. Finally the graph is converted in JSON thanks to the Kgram *toJSON()* function and is shown thanks to the D3 library.

[12]http://d3js.org/

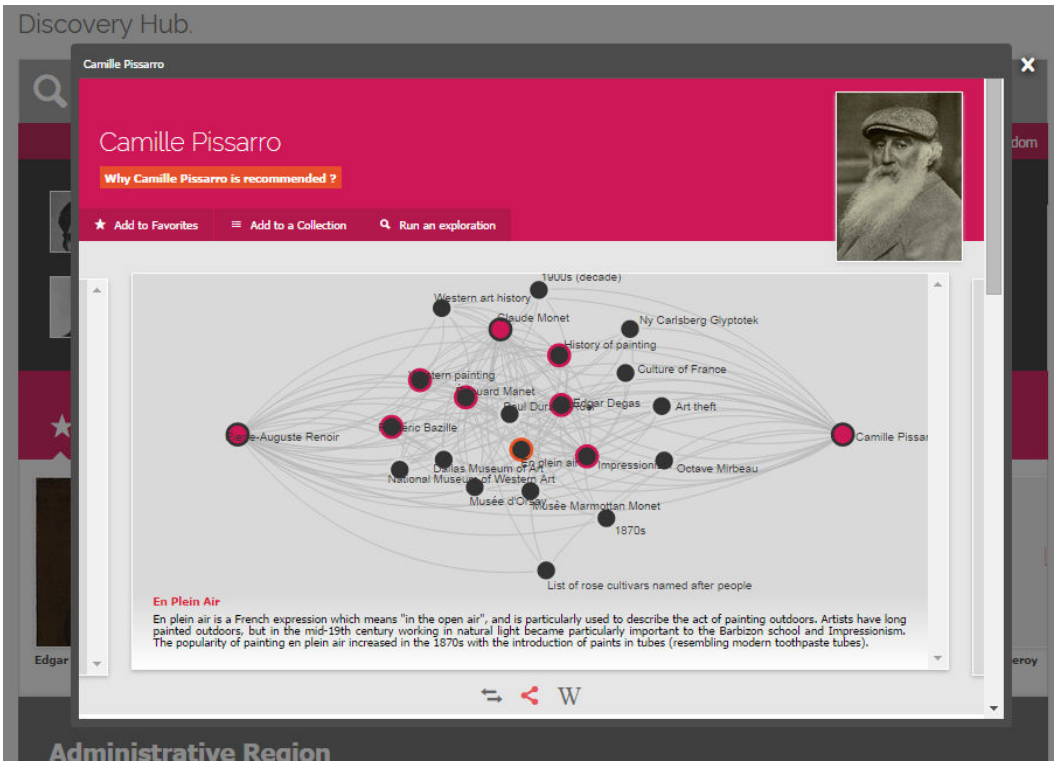

Figure 7.9: Graph explanation

```
CONSTRUCT { ?x ?y ?z } WHERE {
 service <sparqlendpointurl> {
 select distinct ?x ?y ?z where
  { ?x ?y ?z  filter(?x!=?z)
  filter(?x=<dbpedia:Claude_Monet> ||
  ?x=<dbpedia:Pierre-Auguste_Renoir> ||
  ?x=<dbpedia:En_plein_air> ||
  ?x=<dbpedia:Edgar_Degas> ||
  ...
  )
  filter(?z=<dbpedia:Claude_Monet> ||
  ?z=<dbpedia:Pierre-Auguste_Renoir> ||
  ?z=<dbpedia:En_plein_air> ||
  ?z=<dbpedia:Edgar_Degas> ||
  ...
)
}}}
```

Listing 7.2: Example of CONSTRUCT query to build the explanatory graph between the *Claude Monet* and *Pierre-Auguste Renoir* resources

7.2.5 User profile

An important component of the Discovery Hub interface is the user profile which corresponds to the **history** and **extract** components of the information seeking mantra. It is possible to use Discovery Hub without creating an account. Otherwise creating an account allows to benefit from the social mechanisms of the application and to use its memory-features. These functionalities are accessible through the users' profile page, shown on the figure 7.10. There are two possibilities to be registered on Discovery Hub. The first one is to create an account. The second one is to use a Facebook account thanks to the Facebook Connect API[13]. The main advantage of using such social network profile is the possibility to import the Facebook *likes* in Discovery Hub. The likes are matched to DBpedia resources using SPARQL queries that find a correspondence between the two labels (plus few heuristics). The incorrect matches can be selectively removed by the users. The correct *likes* are imported in the users' Discovery Hub favorites.

Creating a user profile allows to access several functionalities and gives a form of persistence that is crucial to distribute the exploration of topics over several sessions:

- The social subscription system (*following*) that gives access to the activity feeds of the other users.
- The favorites that allow to save resources of interest for further re-access or re-use. The favorites are the simplest way to save a resource, they are not structured.
- The collections are used to organize resources that are associated to a user-declared high level interest.
- A history of the searches.

7.3 Discussion: design and redesign rationale

In this part we start by presenting the first version of the Discovery Hub interface in order to explain some design choices of the actual one. Second we discuss its compliance with the design guidelines available in the literature. Third we discuss the correspondence between the interface functionalities and the desired effects of exploratory search systems from chapter 2. In order to structure the reflection we present the actual Discovery Hub task-tree[14], on the figure 7.11:

7.3.1 Discovery Hub V1 design and limits

The Discovery Hub interface and interaction model have significantly benefit from the experience we had with the first version. We present this first version below in

[13] https://www.facebook.com/notes/facebook/facebook-across-the-web/41735647130
[14] modeled by Émilie Palagi

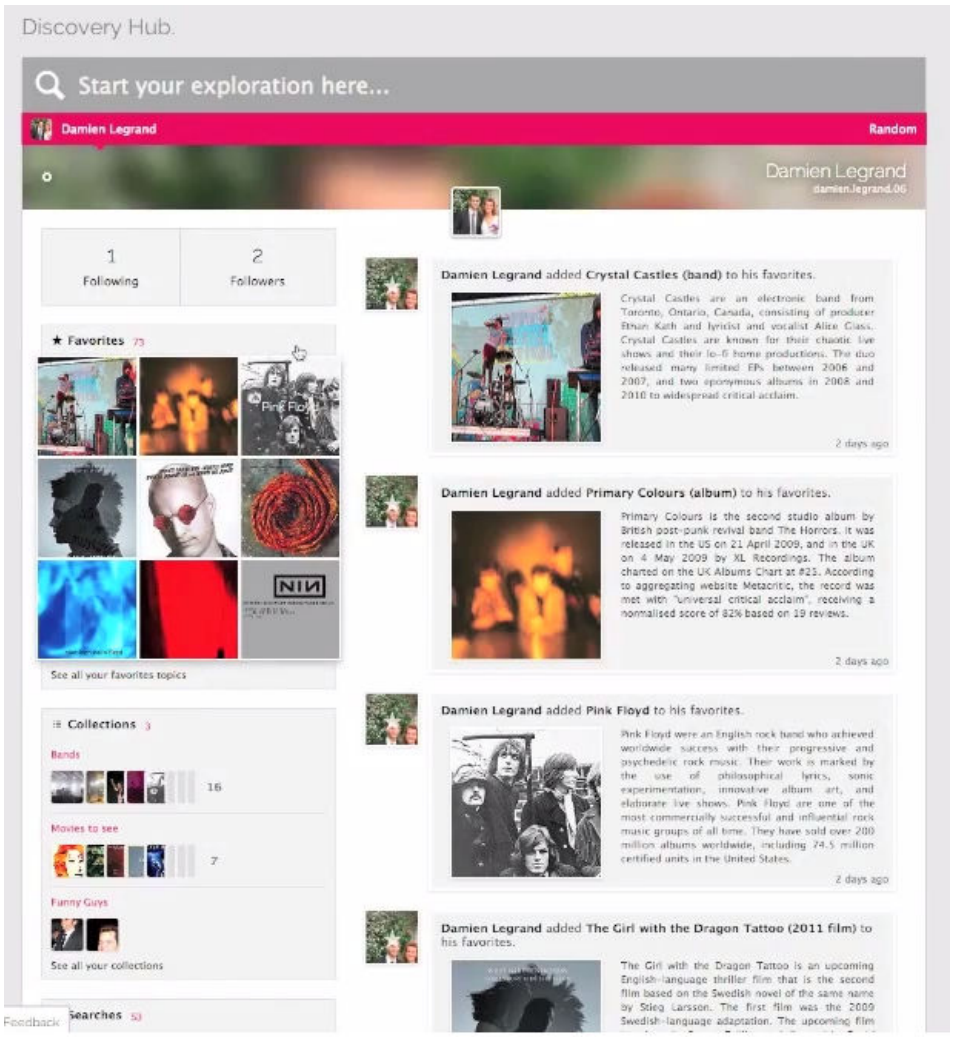

Figure 7.10: The user profile

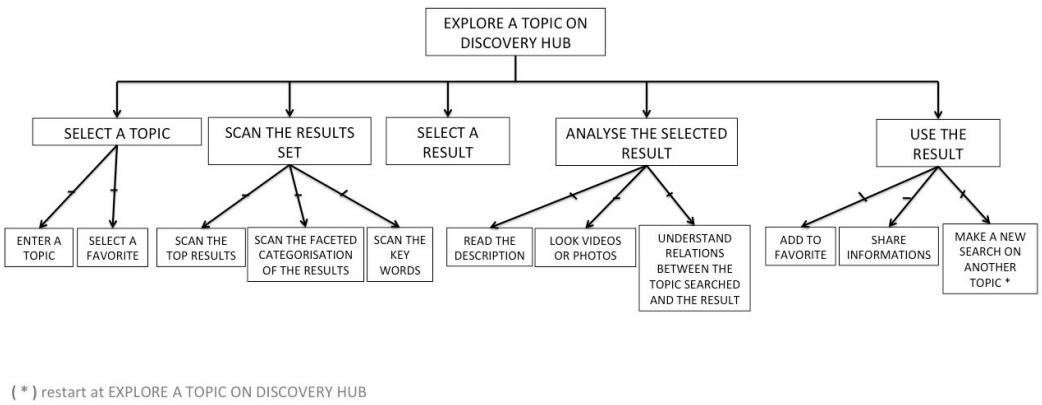

Figure 7.11: Discovery Hub V2 task-tree

order to motivate several strong design and interaction choices we made to conceive the actual application. Several videos showcasing the first version of Discovery Hub are available online[15].

Homepage. The homepage of the V1, shown on the figure 7.12, did not give any explanation about the objective of the tool. There was no tutorial either. The accent was set on the esthetic of the homepage instead of such explanations to motivate the new users. Random topics taken in the query-log were displayed in squares having different sizes. The search bar was also more discrete and there was no call-to-action. To get information about exploratory, semantic search and Discovery Hub in particular the users had previously to go to the *about* section.

Querying. The search bar, shown on the figure 7.13 was minimalistic. Only the label of the resources was shown, giving far less information to the users for its query building. In order to access the advanced search functionalities the users had previously to go to the *query* page. The advanced search mechanisms were not available through the homepage. The composite queries were encouraged thanks to the *Search Box*, shown on the figure 7.14 in which the users drag-and-dropped resources of interest all along their navigation (from profile, results page, etc.). The idea was interesting but led to co-existence of 2 parallel search means. This multiplication of querying features was avoided in the second version which proposes a unique, very visible search bar.

Browsing. The first version of Discovery Hub had 2 result lists views. The first one was called the *mosaic-view* and is visible on the figure 7.15. The second was called the *detailed-view* and is shown on the figure 7.16. The mosaic was a very long results' picture board in which the users had the possibility to scroll rapidly using a faceted browsing system available on the left. It had the objective of giving a quick overview of the results using a very visual layout. The detailed view was conceived to explore deeply the results. It showed the class-facets available (from the CPD) and the categories facets (filters) available for each of them. The explanations were given directly on the *detailed-view* result list page.

The use of 2 result lists was not retained for the conception of the V2 because it increased considerably the complexity of the interaction model and of the interface. Another problem was that the users were forced to switch from the mosaic to the detailed view in order to get explanations about the results. Moreover the display of the unitary results was problematic as explained here-after.

Result page. In the first Discovery Hub version the results were displayed in a separate HTML page, see figure 7.17. It caused an important disruption in the exploratory search process. First the informational context that led to the result, such

[15]https://www.youtube.com/watch?v=MUK01T-n1Ks

as the facet currently opened and the filters applied, was lost. Second the explanations were given on the results list and not on the result page. In other word the result page was a component that was totally independent from the search context. This was corrected in the second version with the display of the results in the form of pop-ups. The pop-ups do not alter the results list interaction state and provide the explanations. The in-pages semantic browsing mode did not include the breadcrumb functionality.

Explanatory features. Explanations were proposed for all the results in the first version. It led to numerous useless clicks that did not retrieve any explanation. This last point was corrected on the actual version of Discovery Hub where only the non-empty explanations are proposed to the users. The common-triples explanation, shown on the figure 7.18, was not displayed in natural language form but in less user-friendly sequences of property-objects. The Wikipedia-based explanation, visible on the figure 7.19 was also slightly different. It directly displayed an HTML Wikipedia page. Such page was not scrollable and users passed from fixed position to another (if several cross-references were found) thanks to arrows on the top-right. The fixed position corresponded to portions of the page where a cross-reference appeared. A limit of this previous implementation was that it was able to show only one page: if a cross-reference appeared in the query-resource Wikipedia page this page was displayed. If it was not the case but a cross-reference appeared in the result-resource Wikipedia page, this page was shown instead. The actual version overtakes this by being totally bidirectional showing all the cross-reference in a compact format (using paragraphs instead of the whole page). Apart from minor esthetic improvements the graph based explanation functionality was not significantly modified, see figure 7.20.

User profile. The user profile is visible on the figure 7.21. It is rudimentary if we compare it to the actual user one. It already supported the Facebook *likes* import, see figure 7.22, as well as the system of favorites internal to Discovery Hub. Nevertheless it lacked all the social mechanisms, the collections and the search history. The user profile is significantly more powerful in the actual Discovery Hub version.

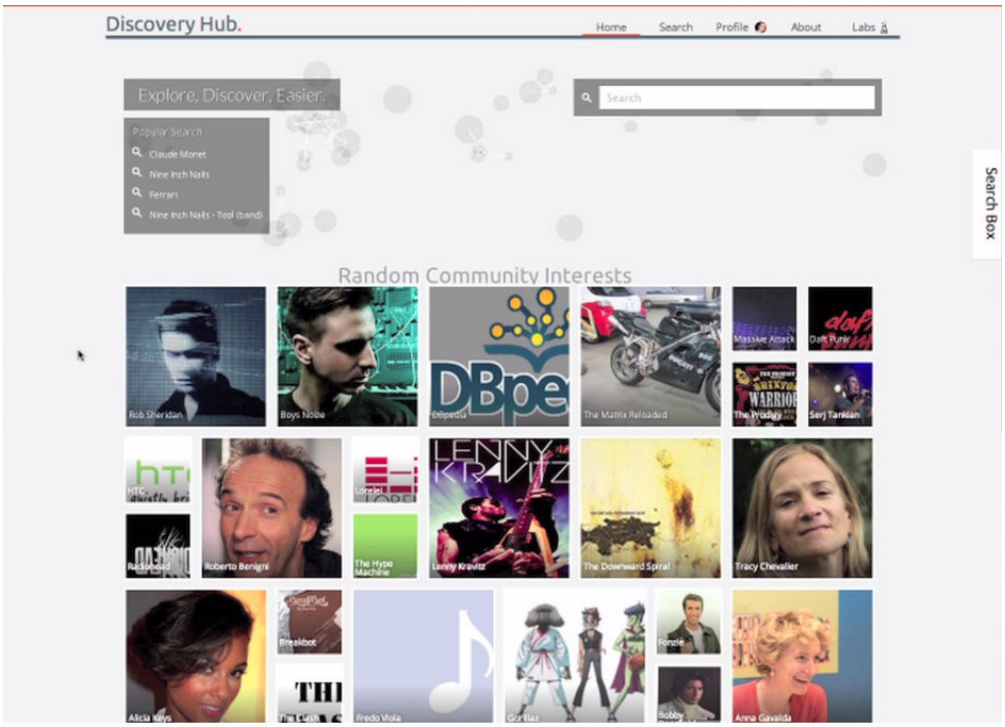

Figure 7.12: Homepage V1

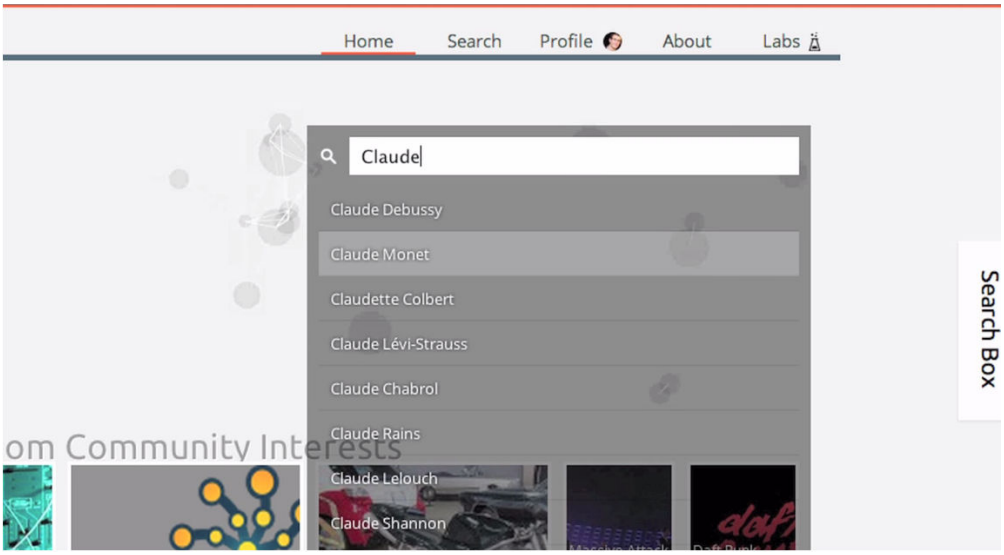

Figure 7.13: Search bar V1

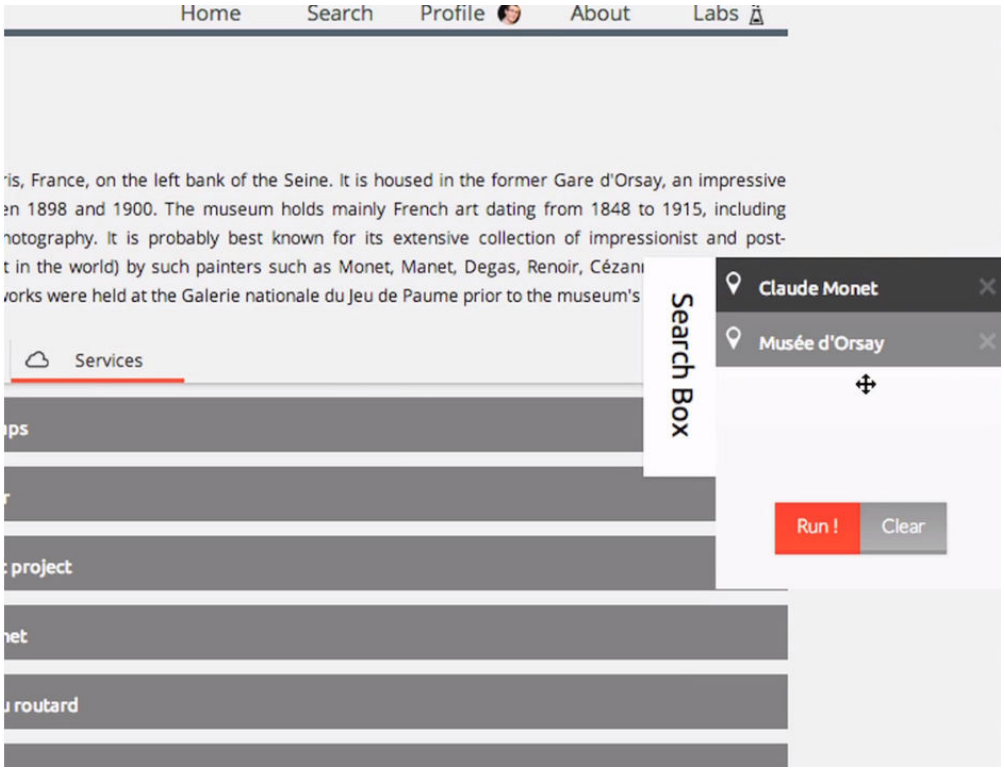

Figure 7.14: Search box V1

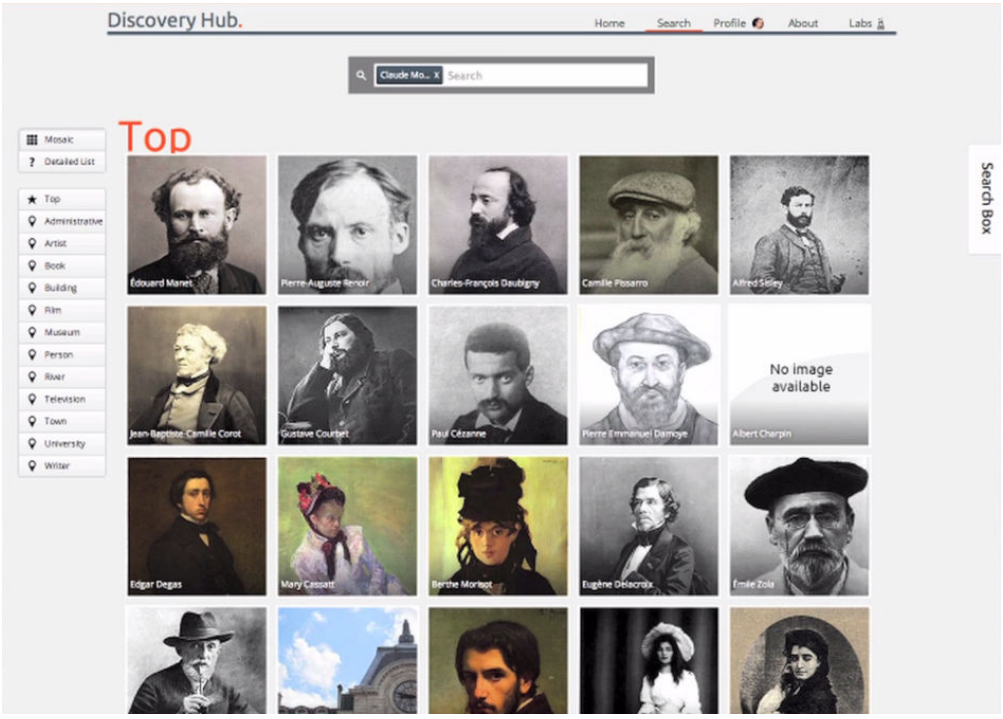

Figure 7.15: Result set *mosaic-view* V1

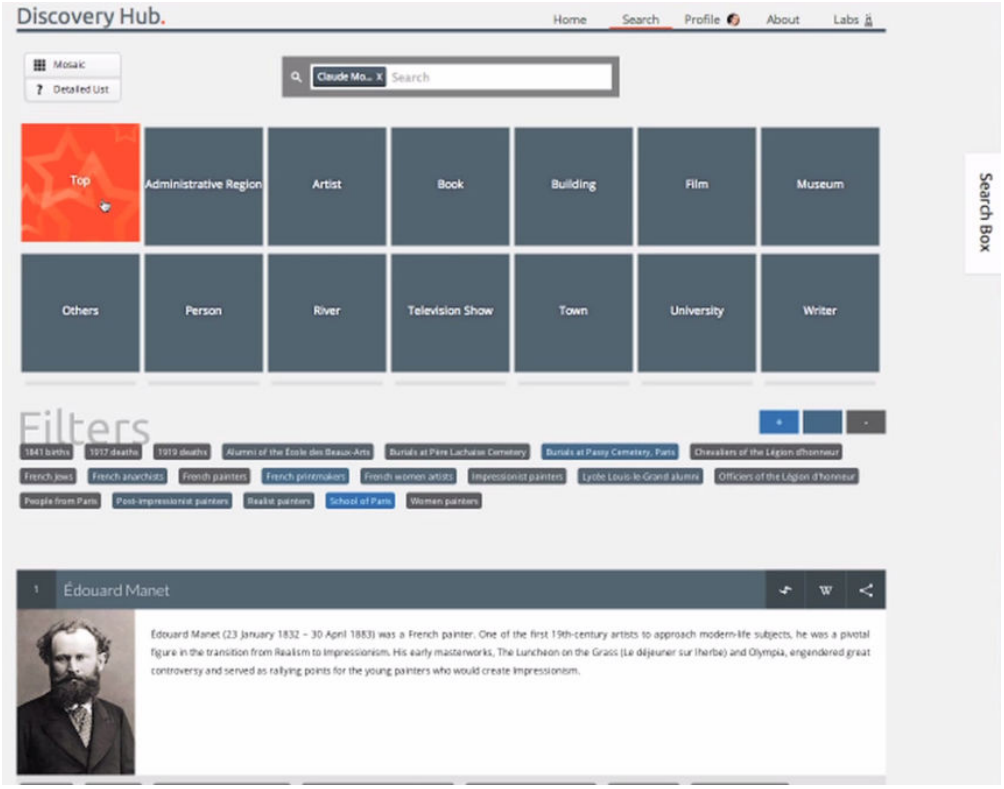

Figure 7.16: Result set *detailed-view* V1

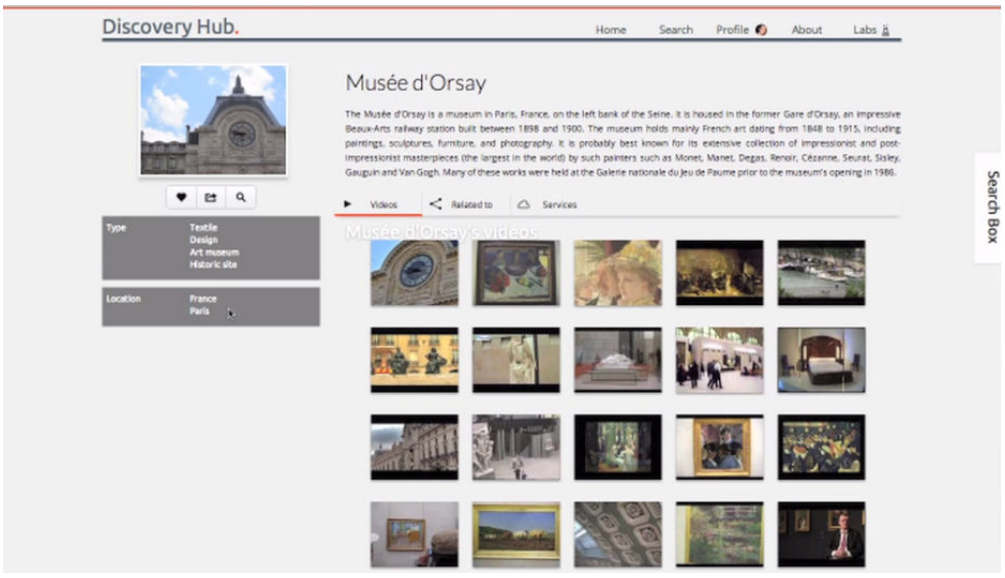

Figure 7.17: Result page V1

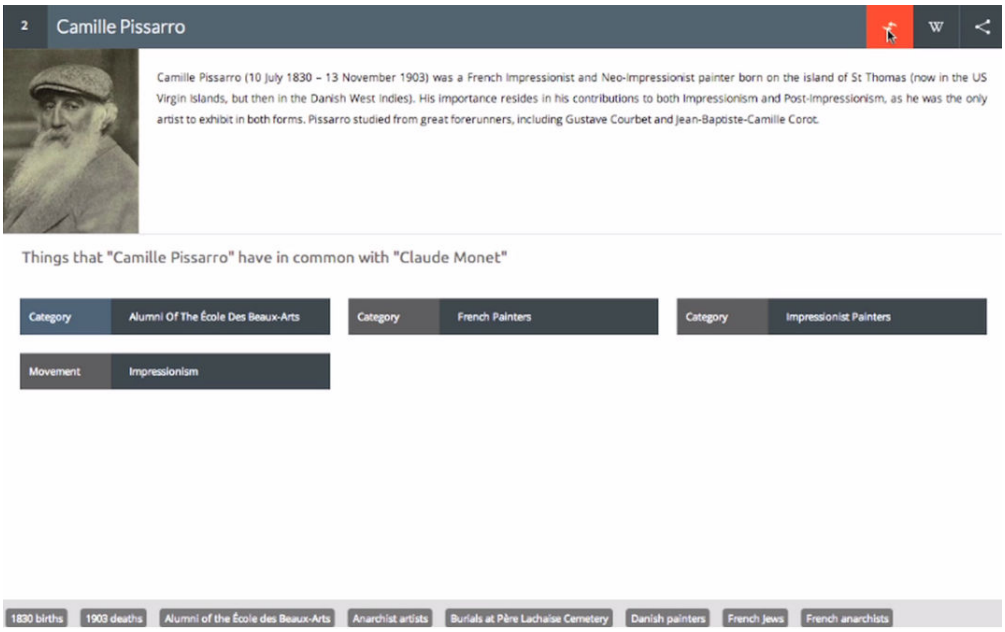

Figure 7.18: Common-triples explanation V1

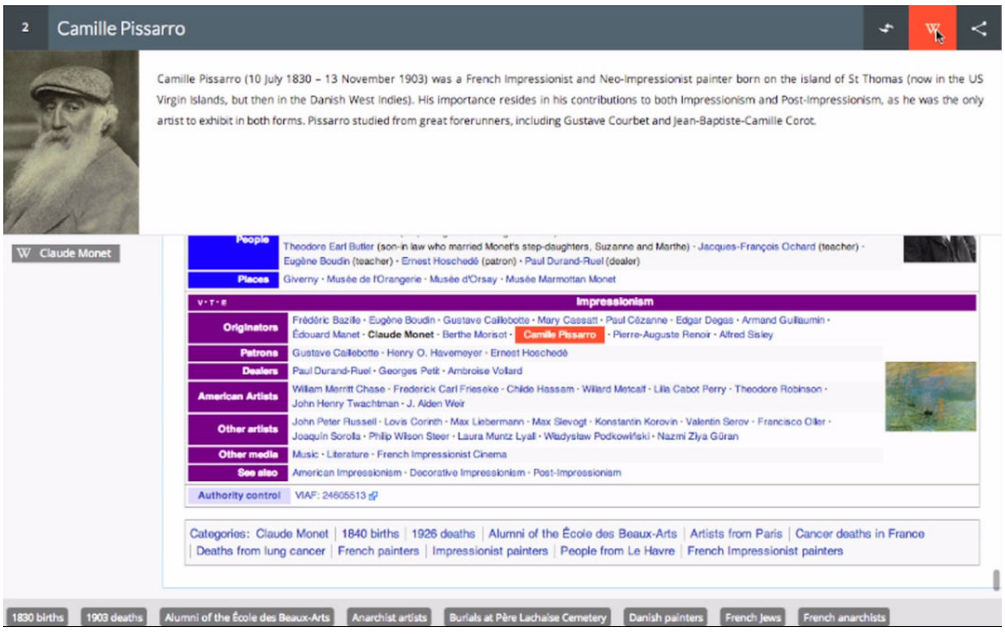

Figure 7.19: The Wikipedia cross-references explanation V1

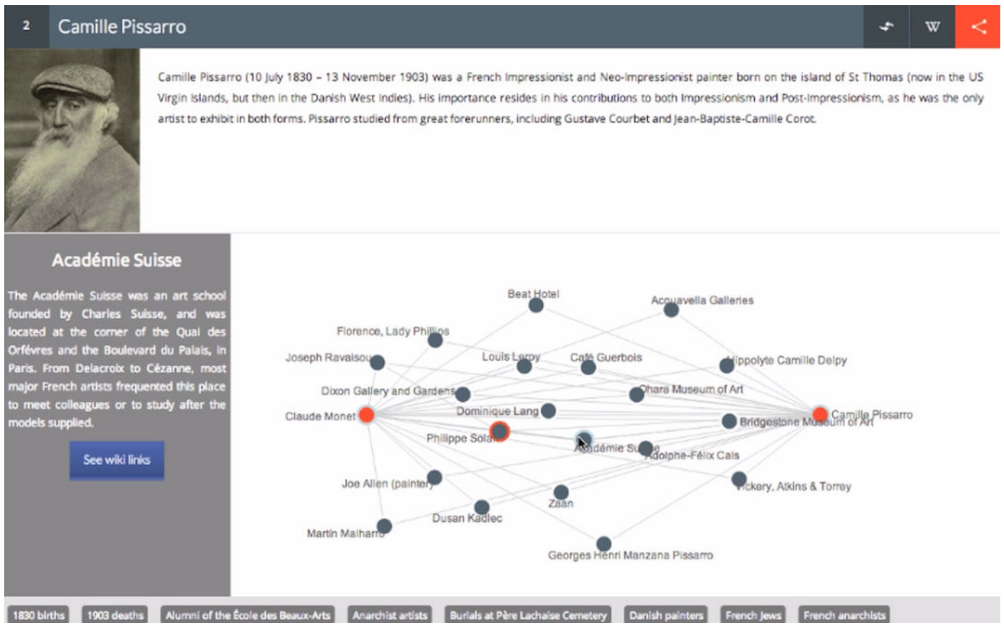

Figure 7.20: Graph-based explanation V1

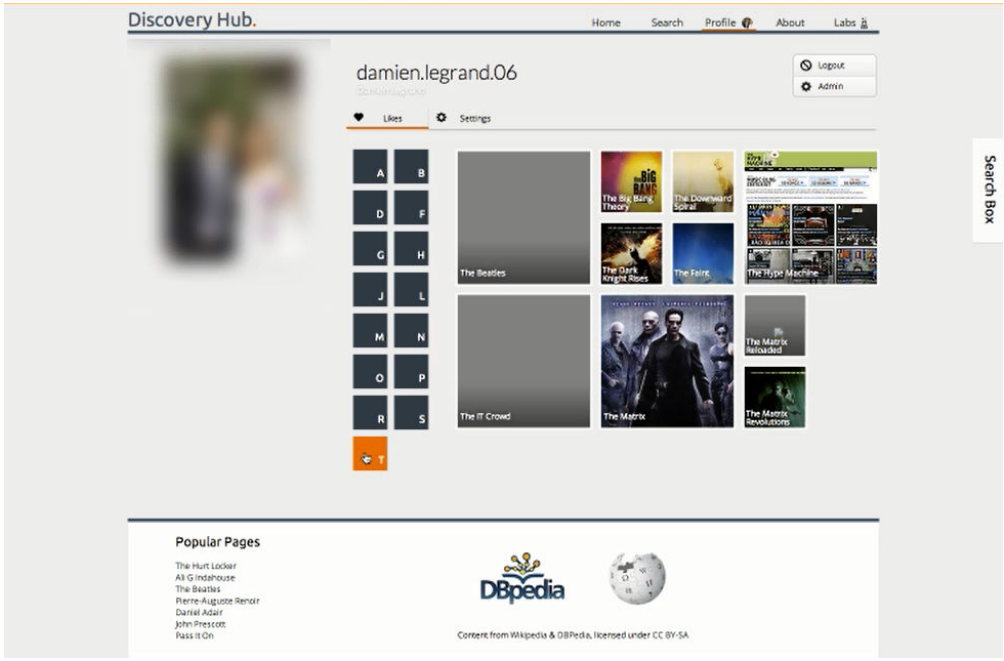

Figure 7.21: User profile V1

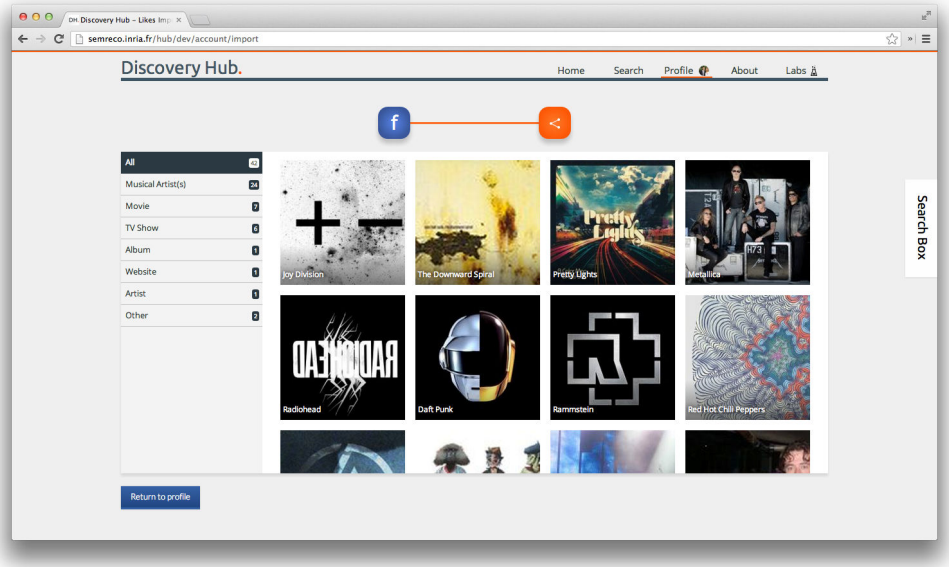

Figure 7.22: Facebook *likes* import V1

7.3.2 Compliance with the existing guidelines

In this subsection we discuss the compliance of the actual interface with the design guidelines identified in the literature (see chapter 2, page 21). We remind that the following guidelines are summarized from [201]:

- **Guideline 1: "*maintain keyword search*"**: the lookup system actually in place on Discovery Hub mimics the keyword search to a certain extent. Otherwise enabling keyword-search in semantic search systems where the resources are the unit of information raises many research questions. For instance the problem of mapping a keyword-query into spreading activation stimulations raises several problems such as establishing the mapping between the keywords and the knowledge graph processed as well as attributing of the stimulation level for each keyword composing the query.
- **Guideline 2:** *"Think about different users"*: in our opinion Discovery Hub is able to support different kinds of users. This statement is notably backed by the use analysis presented in chapter 8, see the appendix I on page 253. The objective of supporting the understanding of various users, potentially having different mental frames was also a motivation for the proposition of different explanatory features.
- **Guideline 3:** *"Show information in context"*: we particularly paid attention to show the information in context. First, the facets are always visible. Second, the result lists are long in order to retrieve elements of context. The result pop-up displays data related to the browsed result in order to contex-

tualize it. The explanations, notably the graph-one, also reveal elements of context.

- **Guideline 4:** *"Include sorting and filtering"*: the faceted mechanisms (browsing and filtering) have this role to a certain extent. Otherwise it could be useful to provide re-ranking and sorting functionalities for a better exploitation of the result lists.
- **Guideline 5:** *"Facilitate information gathering"*: Discovery Hub has several functionalities helping this information gathering such as the Facebook *likes* import, the favorites and the collections.
- **Guideline 6:** *"Offer previews"*: Discovery Hub is not compliant with this guideline, it does not offer any preview mechanism such as numeric values indicators.

7.3.3 Exploratory search desired effects

Only a solid scientific protocol and experimentation can validate the benefits of an interface. This is the purpose of chapter 8. The objective of the preliminary discussion here-after is to highlight the links between the systems desired effects we want to obtain and the application functionalities that aim to produce them.

- **The system provides efficient overviews**: the visual result list and the visibility of the class-propagation domain classes aim to give an overview of the result set.
- **The system shapes the mental representation of the user and its answer-framework**: the dynamic combination of class-based and instance information aims to shape the users mental models by creating a resonance between its mental frame, the schema structure and the instances.
- **The users explore multiple results and browsing paths**: the amount of results retrieved, the results pop-up, the fact of having a traversal browsing system encourage the users to explore various browsing paths.
- **The system inspires the users and shape the information need**: supporting orienteering with several algorithm variants aims to inspire the users and to make them easily discover the information domain without any prior knowledge.
- **The system favors discoveries**: the framework enforce the provocation of discoveries by proposing querying variants that were specifically created to retrieve unexpected results such as the randomized variant or the composite queries. Moreover the advanced querying functionalities aim to unveil hardly identifiable knowledge nuances that can also trigger discoveries.
- **The system eases memorization**: Discovery Hub proposes several in-session (breadcrumb) and account-related memory features.

7.4 Architecture and application design

In this section we present important elements of context about the interface development. After the algorithm implementation the development of the Discovery Hub interface opened a new important chapter of the thesis. It implied work in collaboration, new competencies and some project management concerns. It offered to the author the occasion of supervising two internships including a master degree one: Discovery Hub was developed and demonstrated in constant and close collaboration with the former student Damien Legrand.

7.4.1 Life-cycle of Discovery Hub

The first Discovery Hub version was developed between June and September 2012. The second one was conceived between March and August 2013. The two interfaces were developed with an agile methodology: with constant communication and direct developments. The first version of the system was developed with no solid specifications. It was conceived at its beginning as a demonstrator and a proof-of-concept of the algorithm. However along the inspirations the prototype received more and more functionalities and its interaction model became too complex, see the *discussion* section of this chapter.

The second version of Discovery Hub was re-developed from scratch. Relying on our experience and on the feedbacks we received we did this time a functional modeling using the i* language[16] and some UML diagrams[17], see figure 7.23 and the appendix E. Having a development period of approximately 6 months we also relied on Gantt charts to manage the work load and respect the schedule, see the appendix F on page 245. The objective of the functional modeling was to design a less complex and more intuitive interface. The system affordance was also considered as critical. Several social mechanisms, absent from the first version, were also planned.

7.4.2 Technological and architectural choices

The web application interface is the front-end component of Discovery Hub. It has the role of supporting the users in the exploration of topics. The main technologies used are listed here-after:

- **HTML5, CSS3, JavaScript** (jQuery[18], Backbone.js[19] libraries) for front-end.

[16] *The i* framework proposes an agent-oriented approach to requirements engineering centering on the intentional characteristics of the agent. Agents attribute intentional properties (such as goals, beliefs, abilities, commitments) to each other and reason about strategic relationships. Dependencies between agents give rise to opportunities as well as vulnerabilities. Networks of dependencies are analyzed using a qualitative reasoning approach. Agents consider alternative configurations of dependencies to assess their strategic positioning in a social context* taken from http://www.cs.toronto.edu/km/istar/Overview

[17] http://www.uml.org/

[18] http://jquery.com/

[19] http://backbonejs.org/

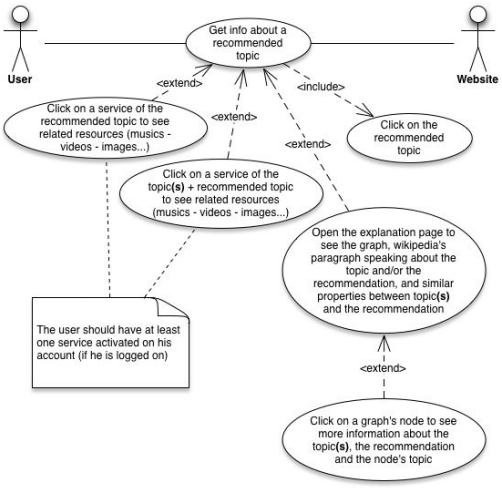

Figure 7.23: An example of UML use-case diagram that served for the Discovery Hub V2 conception

- **PHP** for application server-side with the CodeIgniter framework[20].
- **Virtuoso** was chosen as the database, MySQL was used in the V1. We reused other schema elements such as the SIOC *sioc:UserAccount* and the *sioc:follows* property. Apart from using a single data model and respecting the semantic web standards using Virtuoso offers novel possibilities in term of processing. Indeed the users and the resources they like, search and discover are merged in a unique graph (see figure 7.24). The topics of interest are turned into social objects offering new possibilities in term of querying. For instance it allows to start the spreading activation from a node corresponding to a user or to include people recommendation in the spreading activation results.
- The website is installed on a virtual machine under **Linux Fedora LAMP** configuration.
- **Glassfish**[21] runs the Java algorithm API which was developed using Netbeans[22]. The PHP code communicates with the Java one through command lines. The Java code output JSON formated results.
- **Git**[23] was used for the versioning.

[20]https://ellislab.com/codeigniter
[21]https://glassfish.java.net/fr/
[22]https://netbeans.org/
[23]http://git-scm.com/

Several open-source packages written by Damien Legrand during his internship on the development of Discovery Hub are public and available online:

- **legrand/sparql** for composer/PHP[24]: for generating SPARQL queries in PHP.
- **SPARQL for Bower/JS**[25]: for generating SPARQL queries in JavaScript.
- **SPARQL for Objective-C**[26]: for generating SPARQL queries in Objective-C.
- **legrand/sparqlmodel**[27]: a PHP model interfacing SPARQL endpoints and applications.

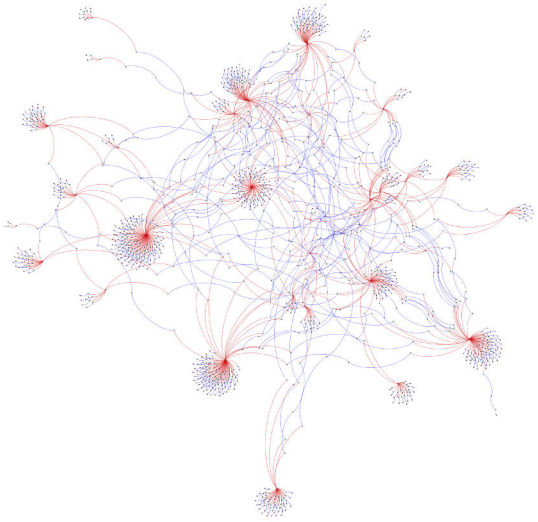

Figure 7.24: A graph visualization of the Discovery Hub database content: the *user-resource* links are shown in red, the resource-resource links are shown in blue

The current hardware configurations of the Discovery application Hub and local DBpedia server are detailed below:

- Application server: 2 processors 32 Go RAM
- SPARQL endpoint: 4 processors 40Go RAM

Run by VMware on 2 Dell PowerEdge R910 physical servers:

- CPU Cores: 16 CPUs x 1.994 GHz
- Processor Type: intel(R) Xeon(R) CPU X7550 @ 2.00GHz
- Processors Sockets: 2
- Cores per Socket: 8
- Memory: 256 GB

[24]https://packagist.org/packages/legrand/sparql
[25]https://github.com/snoozeman/sparql-js
[26]https://github.com/snoozeman/sparql-objectivec
[27]https://packagist.org/packages/legrand/sparqlmodel

7.4.3 Communication

We communicate about the application outside the scientific community. The main objective was to increase the amount of queries processed in order to obtain a longer query-log i.e. usable for further researches. A Twitter account was opened (discovery_hub[28]). The application was also demonstrated at the entrance booth of the 2013 Bell Labs France open days[29]. Last but not least a commercial adaptation of the algorithms for web applications cold-start personalization (using social insights like the Facebook likes) won the 2013 *Challenge Jeunes Pousses* entrepreneurial challenge[30]. This challenge is organized every year by the Sophia-Antipolis Telecom Valley innovation cluster[31]. The adapted project was named *The Showcase Machine*, see the appendix G on page 247.

7.5 Conclusion

In this chapter we presented the Discovery Hub platform, architecture, technologies, interface and interaction model. We did not focus strictly on the application but also gave some elements of context and discussion. First we detailed the interface and its principal component: the homepage, querying system, results list, result page, explanation features and user profile. Second we discussed the current interface and interaction model by explaining the important design choices in regard of our experience with the first version of the application. We also put it in regard of available guidelines for exploratory search systems and proposed a correspondence between the desired effects of an exploratory search system and the Discovery Hub functionalities. Finally we detailed the context of the development of the web application in terms of project management, technologies and communication. We need now to evaluate the relevance of the whole: algorithm and interface. The following chapter is dedicated to such evaluations.

[28] https://twitter.com/discovery_hub

[29] http://www.alcatel-lucent.com/fr/blog/corporate/2013/07/les-open-days-levenement-incontournable-de-lannee

[30] http://www.webtimemedias.com/article/challenge-jeunes-pousses-vainqueur-showcase-machine-20131007-53221

[31] http://www.telecom-valley.fr/

CHAPTER 8

Evaluating Discovery Hub

Contents

8.1 Introduction

In this chapter we present the evaluations[1] we did for the monocentric, polycentric algorithms as well as for the criteria of interest specification and the controlled randomness injection variants. Such evaluations aim to verify that the algorithms retrieve results that are relevant in an exploratory search context. To this end we did not only focused on their relevance but also on their novelty and unexpectedness. Indeed a successful exploratory search engines should provoke multiple discoveries about the topic explored.

[1]The author especially thanks Émilie Palagi, former master's degree intern co-supervised by the author, Alain Giboin (INRIA), Gessica Puri (INRIA) and Florentin Rodio (Bell Labs) for their precious contributions to the evaluations

As mentioned in chapter 2 the evaluation of the exploratory search systems is especially challenging and is still today an open challenge. The research needs to be pursued to reach a community agreement about the best protocols and methodologies to properly evaluate such systems. The publications in the literature often extensively motivate and explain the design of the evaluation protocols they use.

Discovery Hub is an algorithm-based exploratory search engine having the specificity of implementing multiple algorithms. The corresponding query-modes can be used to deeply explore a topic, through numerous angles by extensively relying on the result sets that are automatically identified for them. Thus the majority of the evaluations were focused on the users' perception about the results retrieved. Three distinct evaluations, that were all based on human judgment, were executed. The first was dedicated to the mono-centric queries, the second to the polycentric ones and the last one for the 2 advanced querying variants. Finally we present in this chapter an ongoing reflection about an evaluation protocol aiming to overtake some limits of our previous evaluations and of the evaluations published in the literature.

In this chapter we will review (8a) the mono-centric queries evaluation protocol, experimentation and results (8b) the poly-centric queries evaluation protocol, experimentation and results, (8c) the criteria of interest specification and controlled randomness injection algorithm variants the protocol, experimentation and results (8d) we present an ongoing reflection about a novel protocol for the evaluation of exploratory search system using some innovative approaches (8e) we also reveal some usages statistics computed from the Discovery Hub query-log. The evaluations of the monocentric, polycentric and advanded querying variants were respectively published in [128], [123] and [126].

8.2 Monocentric queries evaluation

This section describes the first evaluation which was done to verify the results relevance. We evaluated the most basic (but the most frequent) queries possible with the framework: the mono-centric ones. This first evaluation occurred in two rounds:

- First we evaluate the results against a baseline on a neutral interface (at the time the Discovery Hub web application did not exist yet).
- Second a fraction of poorly rated results were judged again through the Discovery Hub interface, using the explanatory features.

8.2.1 Protocol

We evaluated the results of our monocentric semantic spreading activation algorithm (mentioned as *MSSA* hereafter) against the sVSM algorithm used in the MORE recommender [133][2]. The main reason of this choice is that MORE was

[2]The author thanks Roberto Mirizzi for his precious help concerning the MORE API.

at the time the only similar linked-data based system that has been compared to another: Seevl, see [43]. As the purpose of MORE is movie recommendation only the *Film* results facet of our framework was taken into account during the comparison. Comparing our results to the MORE ones was acceptable at this point for the following reasons:

- As mentioned in the dicussion section of chapter 5, similarity is an important factor of relevance for us, see the section 5.7 page 129.
- Even if the major added value of Discovery Hub is to find results which are not obvious, the results that are similar to the topic(s) queried play an important role as they considerably reinforce the users' confidence in the system.
- In our framework the activation value of the resources similar to the topic explored are reinforced. They are often very well-ranked.
- More generally there is a synergy between the recommenders and exploratory search systems that was already discussed in chapter 4, see the section 4.3 page 79.
- More generally there is a synergy between the recommenders and exploratory search systems that was already discussed in chapter 4, see the section 4.3 page 79.
- The MORE recommender compared its results to another approach, it is important to encourage such comparison between the systems.

During the first round of experimentation we evaluated both the relevance and the discovery potential of our results regarding the sVSM baseline. We formulated the following hypotheses:

- **Hypothesis 1:** The MSSA algorithm gives results at least as relevant as the sVSM one, even if it is not domain-optimized (the implementation in the MORE application is specific to the cinema-domain).
- **Hypothesis 2:** The MSSA algorithm has less degradation than the sVSM algorithm. In other words, its end-list results are judged as better as the one of sVSM.
- **Hypothesis 3:** There is a greater chance that the results are less relevant but novel to users at the end of the lists.
- **Hypothesis 4**: The explanatory features increase the users' overall judgments positivity.

The hypotheses 1 and 2 aim to verify that the MSSA algorithm retrieves relevant results compared to the domain-specific implementation of sVSM for the MORE movie recommendation use case. The hypothesis 3 aims to verify that the algorithm correctly ranks the results by retrieving the most relevant ones first. Presenting the most relevant and less novel results in priority increases the users confidence in the system. However in an exploratory search context it is important to

present less-known, novel results to the users as they want to increase their knowledge on the topic searched. The hypothesis 4 aims to evaluate the influence of the explanatory features on the users' results perception.

The participants evaluated alone the algorithms results on a neutral interface set up with the online survey solution Limesurvey[3]. They had to judge 5 lists of movies' recommendations. These lists were composed of the top 20 results from the 2 algorithms. Each list was generated starting from one *seed-film*. The lists were fully randomized in a single list and doubloons were removed. Thus the participants were not aware of the results provenance. The seed-films used to generate the lists were randomly chosen in the "*50 films to see before you die*" list[4]. It was chosen because of its diversity: *"each film was chosen as a paragon of a particular genre or style"*[5]. The randomly selected seed-films were: *2001: a space odyssey, Erin Brockovich, Terminator 2: judgment day, Princess Mononoke and Fight club*. Two Likert scale [112] questions were asked for each result. The first to evaluate the similarity, the second to evaluate the novelty:

- Question 1: *With the film 2001: a space odyssey, I think I will live a similar cinematic experience as with planet of the apes? Strongly agree, agree, disagree, strongly disagree*
- Question 2: *You and 2001: a space odyssey? Seen, Known but not seen, Not known*

In order to analyze the relevance and the discovery potential a 2 (MSSA vs sVSM) * 5 (Film 1 vs Film 2 vs Film 3 vs Film 4 vs Film 5) * 2 (1-10 ranks vs 11-20 ranks) analysis of variance (ANOVA) test was realized. In statistics, an ANOVA [171] is a method used to compare more than two means simultaneously and determine if their differences are substantial and reflect natural sampling fluctuations. It tests which proportion of the total variance in the data can be be attributed to the experimental factors, and whether this deviates from what could be expected if the variance is due to noise. As we study several factors at the same time and as the users participate in all conditions, we performed a factorial ANOVA with repeated measure. The survey was filled by 15 persons resulting in a total of 3750 evaluations. The participants sample was composed of 2 females, 13 males, with an average age of 31.7 years, being mainly computer scientists. The average number of movies seen by month on any support was 10.4 (standard deviation $sd = 8.66$).

In order to analyze the impact of the explanatory features on the users' perception of the results, the participants of the first experimentation were asked later to evaluate again 20 results through the Discovery Hub interface. The explanations helped them to make a choice this time. The 20 results were randomly selected in the MSSA results list of the first evaluation. We constrained the random selection to poorly evaluated results (*disagree, strongly disagree* answers to the question one) in order to observe an eventual improvement.

[3] www.limesurvey.org/

[4] http://www.film4.com/special-features/top-lists/top-50-films-to-see-before-you-die

[5] http://en.wikipedia.org/wiki/50_Films_to_See_Before_You_Die

Measure	Algorithm	Rank	Mean	Standard Deviation
Relevance	MSSA	1-10	1.54	0.305
		11-20	1.28	0.243
	sVSM	1-10	1.42	0.294
		11-20	0.93	0.228
Discovery	MSSA	1-10	1.1	0.247
		11-20	1.21	0.228
	sVSM	1-10	1.14	0.251
		11-20	1.5	0.205

Table 8.1: Scores for partial lists, monocentric evaluations

8.2.2 Results

In the following results 0 corresponds to *strongly disagree*, 1 to *disagree*, 2 to *agree*, 3 to *strongly agree* for the *relevance score*. 0 corresponds to *seen*, 1 to *known but not seen*, 2 to *not known* for the *discovery score*. The score for each user was very stable because it was computed over a large number of responses (250 per user) and over two major sources of variation (ranking and film). Having a large amount of evaluations per user increases the reliability on measurement setting and thus has a positive impact on the power of the statistical testing [22].

Hypothesis 1. In order to verify hypothesis 1, we observed the difference between the MSSA and the sVSM relevance scores. The figure 8.1 shows that overall MSSA (mean $m = 1.42$, standard deviation $sd = 0.27$) outperforms sVSM ($m = 1.18, sd = 0.24$). The ANOVA test being statistically significant ($F(1, 14) = 113.85$, $p < .001$) the hypothesis 1 is verified. The full ANOVA results are available in appendix H on page 249.

Hypothesis 2. In order to verify hypothesis 2, we observed the difference between the MSSA and the sVSM relevance scores at the end of the top results list (rank 11-20). The table 2 presents the average scores of relevance and discovery for the beginning and the end of result lists. SSA has a better relevance score ($m = 1.28, sd = 0.243$) than sVSM ($m = 0.93, sd = 0.228$) for the results at the end of the list. The ANOVA test being statistically significant ($F(1, 14) = 20.23, p = .001$) the hypothesis 2 is validated.

Hypothesis 3. In order to validate the hypothesis 3 we compared both the relevance and discovery scores of the 2 two algorithms for the beginning and the end of the result lists. The results are perceived less relevant in the second half of the list (beginning $m = 1.48, sd = 0.299$, end $m = 1.10, sd = 0.235$) but have a higher discovery score (beginning $m = 1.12, sd = 0.249$ vs end $m = 1.355, sd = 0.216$). The ANOVA test being statistically significant for relevance ($F(1, 14) = 134.02, p < .001$) and discovery ($F(1, 14) = 64.30, p < .001$), thus the hypothesis 3 is validated. It is noticeable that sVSM has a better discovery score than MSSA at the end of the list but at the same time its relevance decreases considerably. The MSSA algorithm can be considered as more balanced.

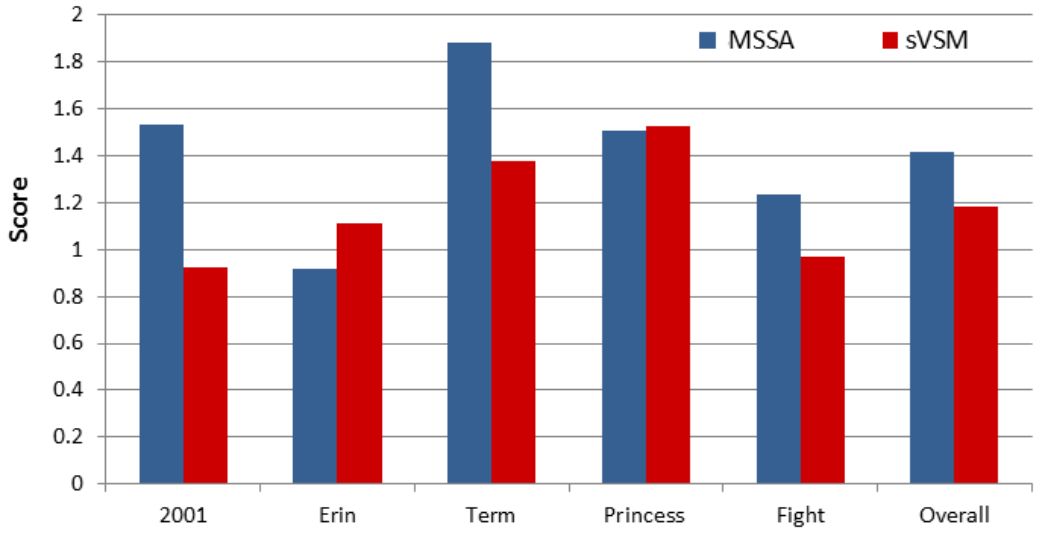

Figure 8.1: Relevance and discovery scores for each seed-film

Hypothesis 4. In order to verify hypothesis 4, we observed the difference between the relevance scores obtained with and without the explanatory features. The relevance score rose significantly: previously $m = 1.26$, $sd = 0.40$, with the features: $m = 1.50$, $sd = 0.26$. The average number of positive judgments reached 9.4 versus 7.34 previously. A Student test [171] was performed. It is used instead of the ANOVA when only two means are compared. The Student test being statistically significant ($t(14) = 3.872$), $p = 0.002$) the hypothesis 4 is verified: the explanations features enhance the perception of the results.

We also asked the participants to give a qualitative feedback about the three explanatory features. For the 3 features we asked "*the feature helped me to understand the relation between the movies and to make a choice?*" and one more general question: "*overall, I feel that these three features can help me to make new discoveries*". 0 corresponded to *strongly disagree*, 1 to *disagree*, 2 to *agree* and 3 to *strongly agree*.

According to the users the common properties and the graph-based features helped significantly the participants (average scores: 2.13 for both) whereas the benefit of the Wikipedia-based feature was less evident (average score: 1.86), see figure 8.2. The more general question received the high average score of 2.53. However the results are not uniform and show the interest to propose different explanatory features. To conclude such functionalities have a positive impact on the users' perception of the results. They increase their confidence and make their exploration easier at the same time.

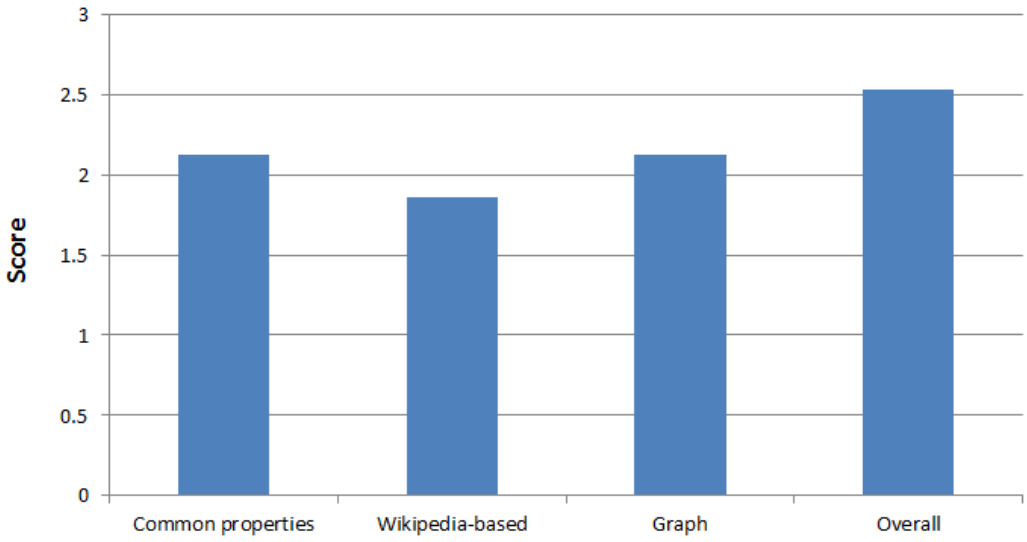

Figure 8.2: Users' opinion about the explanatory features

8.3 Polycentric queries evaluation

8.3.1 Protocol

We did not find a relevant baseline for the composite query results comparison. We formulated the following hypotheses:

- **Hypothesis 5**: The composite query results are interesting to the users.
- **Hypothesis 6**: A consequent portion of the results are unexpected; they have a high discovery potential.
- **Hypothesis 7**: The explanatory features help the users to understand the relation between the query-resources and the results; they support efficiently the results space understanding.

This evaluation was executed using the Discovery Hub interface. The users had to evaluate 2 result lists of 10 results. Each list was generated starting from 2 of their individual Facebook *likes* that were randomly combined. In this way we wanted to simulate real composite interest queries that the users were susceptible to enter in the system. The following scenario was introduced: "*you heard about a new discovery engine that can help you to discover new items easily, starting from items you already like. This tool notably allows to generate results starting from several interests. You decide to test it on yours. We propose you to judge 2 result lists generated from your Facebook likes*". The survey was filled by 12 persons: 3 females, 9 males from various backgrounds, mainly people who asked an early access to the Discovery Hub beta. Two Likert scales questions were asked for each result. The first one aimed to evaluate the users' interest, the second one to evaluate their surprise:

- Question 1: *The result interests me: strongly agree, agree, disagree, strongly disagree.*
- *Question 2: The result is unexpected: strongly agree, agree, disagree, strongly disagree.*

8.3.2 Results

In the results presented hereafter 0 corresponds to *strongly disagree*, 1 to *disagree*, 2 to *agree*, 3 to *strongly agree* for both the relevance and the discovery scores (corresponding respectively to questions 1 and 2).

Hypothesis 5. In order to verify hypothesis 5, we observed the relevance score. The average relevance score was 1.65, with a standard deviation of 0.94. The figure 8.3 is the histogram of the average relevance scores per query. 71% of queries received a relevance score over the mean (1.5). Thus the hypothesis 5 is verified. It is noticeable that one case received the worst score possible; all its results were rated 0. The explanation is that the seeds composing the query were very distant: *Samuel L. Jackson* and the music streaming application *Grooveshark*[6].

Hypothesis 6. In order to verify hypothesis 6, we observed the unexpectedness score (question 2). The average unexpectedness score was 1.90 with a standard deviation of 1. The figure 8.4 shows the average unexpectedness scores histogram. 58.33% of queries received an average score over the mean (1.5). Thus, the hypothesis 6 is verified.

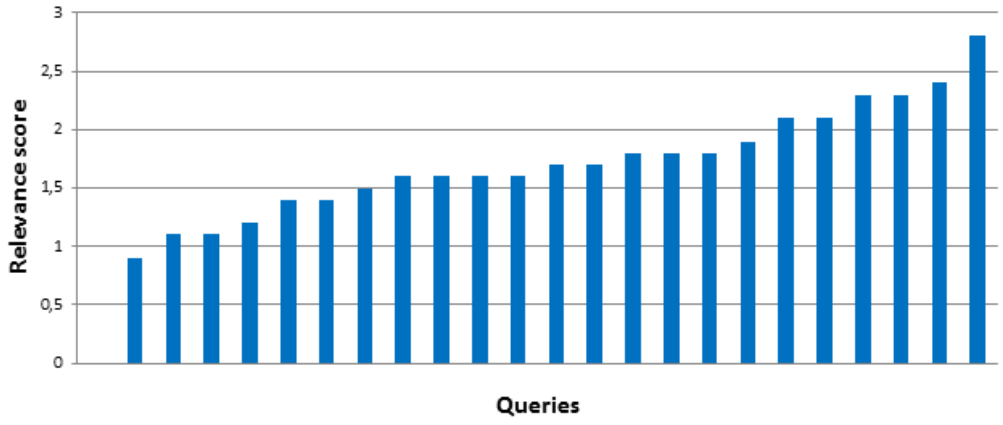

Figure 8.3: Histogram of the relevance scores, polycentric queries evaluated

It is also interesting to observe the recovery between the relevance and the unexpectedness:

- 61.6% of the results were rated as strongly relevant or relevant by the participants.

[6]http://grooveshark.com/

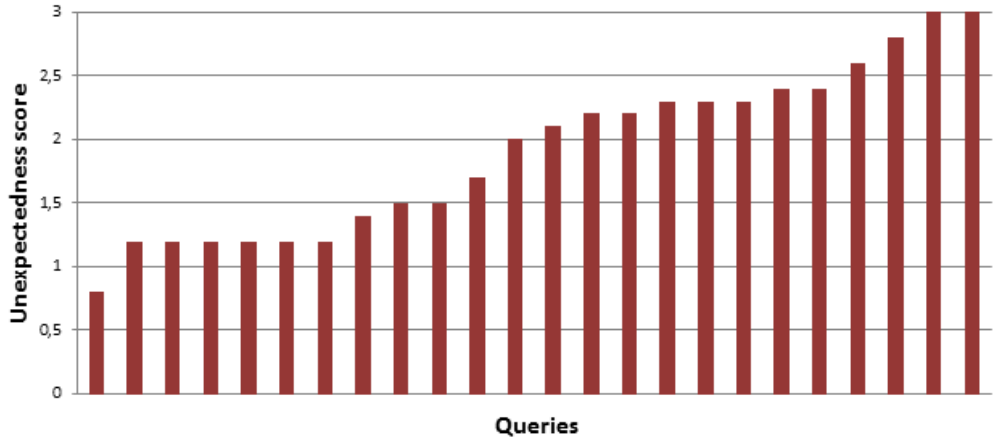

Figure 8.4: Histogram of the unexpectedness scores, polycentric queries

- 65% of the results were rated as strongly unexpected or unexpected.
- 35.42% of the results were both rated as relevant and unexpected. These results are the most valuable but also the most difficult to retrieve as they are often non-obvious.

Hypothesis 7. During the experimentation we also asked to the participants to give their opinion about the three explanatory features. We asked the same question as during the monocentric queries evaluation i.e. "*the feature helped me to understand the relation between the movies and to make a choice?*" and "*overall, I feel that these three features can help me to make new discoveries*". The results are shown on the figure 8.5. The graph-based explanatory feature, which was designed specifically to understand the non-trivial connections between several resources, received a very high average helpfulness score (mean $m = 2.92$). It is particularly adapted to explain the polycentric query results as it shows multiple paths at the crossroad of the result and the seeds. The Wikipedia-based explanatory feature received an average score over the mean ($m = 1.83$). Finally the common property feature received an average score close the mean ($m = 1.58$). It is often impossible to find common triples between the results and all the different seed nodes constituting the composite interest. This feature is more helpful for monocentric queries. The more general question received the high average score of 2.67 and confirms the interest of offering such explanations during composite interest exploration. Regarding all these results the hypothesis 7 is verified. It is also interesting to observe on the figure 8.5 that the users' perception about the explanations helpfulness consequently varies according to the type of query considered (monocentric or polycentric). These results show the interest of offering several distinct forms of explanations in order to cover different information needs.

Finally we asked the participants to rank the 3 functionalities regarding their perceived efficiency in terms of results explanations, see the results on the figure 8.6. The rankings confirmed the previous results. The common property feature

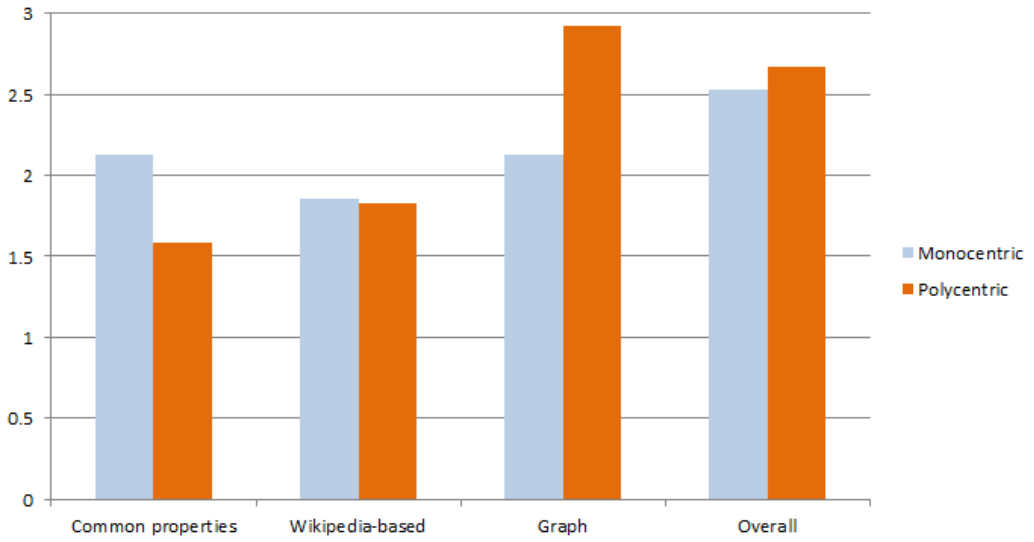

Figure 8.5: Explanatory functionalities perceived helpfulness

was perceived as the less efficient (ranked first: 0%, second: 72.7%, third: 27.3%). The Wikipedia-based feature was more appreciated (54.5%, 27.3%, and 18.2%). Finally the graph-based one received a very large approval (45.5%, 45.5%, and 9%). Nevertheless, the results are not totally homogeneous among the users and confirm again the interest to propose various explanation features.

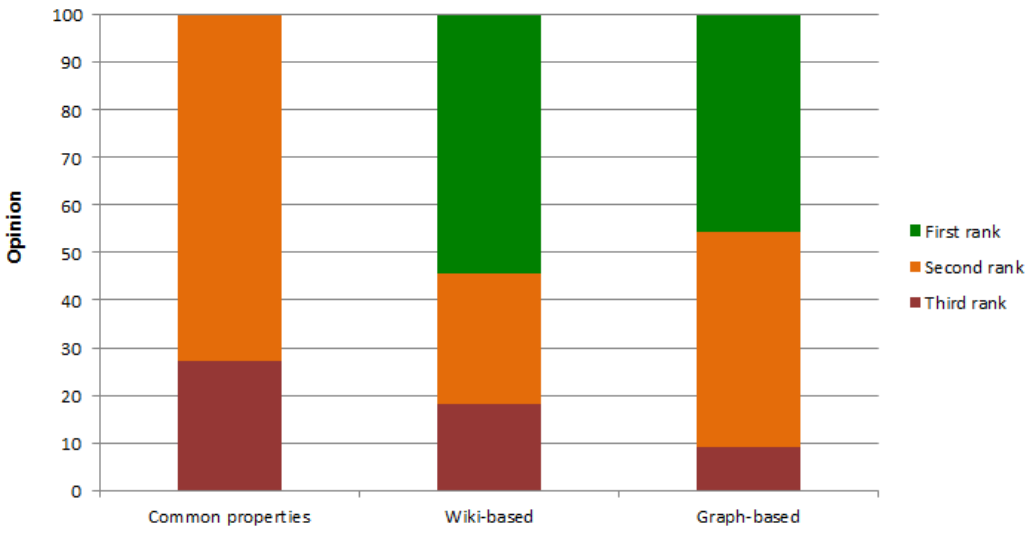

Figure 8.6: Explanatory functionalities ranked by participants perceived helpfulness

8.4 Advanced querying functionalities evaluations

The knowledge nuances we introduced in chapter 5 are quantified below thanks to human evaluation. The boost in terms of relevance and surprise they should provoke is discussed. Two algorithm variants aim to retrieve such hardly identifiable knowledge nuances: the criteria of interest specification and the controlled randomness injection ones.

8.4.1 Preliminary study

Before describing the core protocol dedicated to the advanced querying modes we detail a short preliminary experimentation that was conducted right after the monocentric queries evaluation. The objective was to study the impact of the criteria of interest specification on the results' relevance. We asked to several participants of the monocentric queries evaluation to choose the 3 most interesting aspects (DBpedia categories) and the 3 less interesting ones about the movie Fight Club. Fight Club was chosen because it was the most viewed among the 5 movies (by 86.66% of the participants). Ten persons participated to this experimentation addendum. They showed interest in narrow categories, e.g. *American black comedy films*, and disinterested in broad ones e.g. *1999 films*. The specified criteria of interest and disinterest were used to compute the results with $v = -1$ for the categories specified as uninteresting and $v = 1$ for the ones specified as interesting. The average relevance score of the top 10 results significantly rose compared to the basis algorithm ones: 1.94 ($sd = 0.55$) versus 1.42 ($sd = 0.39$) previously. This encouraged us to pursue the research about the criteria of interest specification variant and to implement it on the online application. Before describing the core protocol dedicated to the advanced querying modes we detail a short preliminary experimentation that was conducted right after the monocentric queries evaluation. The objective was to study the impact of the criteria of interest specification on the results' relevance. We asked to several participants of the monocentric queries evaluation to choose the 3 most interesting aspects (DBpedia categories) and the 3 less interesting ones about the movie Fight Club. Fight Club was chosen because it was the most viewed among the 5 movies (by 86.66% of the participants). Ten persons participated to this experimentation addendum. They showed interest in narrow categories, e.g. *American black comedy films*, and disinterested in broad ones e.g. *1999 films*. The specified criteria of interest and disinterest were used to compute the results with $v = -1$ for the categories specified as uninteresting and $v = 1$ for the ones specified as interesting. The average relevance score of the top 10 results significantly rose compared to the basis algorithm ones: 1.94 ($sd = 0.55$) versus 1.42 ($sd = 0.39$) previously. This encouraged us to pursue the research about the criteria of interest specification variant and to implement it on the online application.

8.4.2 Protocol

In order to confirm the value of such advanced querying modes we built a more solid protocol, presented here-after. This protocol also covers the evaluation of the controlled randomness injection. We formulated the following hypotheses concerning the criteria of interest specification variant:

- **Hypothesis 8**: The users who specify their criteria (categories) of interest before launching the search, find the results of the search more relevant than users who did not specify their criteria.
- **Hypothesis 9**: The users who specify their criteria (categories) of interest do not find the results of the search less novel than users who do not specify their criteria. In other words, there is no loss of discovery power due to the specification of interest criteria.

We formulated the other 2 hypotheses related to the controlled randomness injection functionality.

- **Hypothesis 10**: The stronger is the level of randomness the more surprising the results are for the users.
- **Hypothesis 11**: Even if the level of surprise is high, the majority of the top results are still relevant to the users.

Then we selected the topics used as query-resources for this experimentation. First we randomly chose a set of 20 queries, i.e. DBpedia resources, from the Discovery Hub query-log. These queries are hereafter referred to as *exploration topics*. Second we asked 16 participants to select in this list the 4 topics that were the most interesting to them. We retained the 2 topics selected by the largest number of participants: information visualization[7] and the singer Serge Gainsbourg[8]. This selection is interesting regarding exploratory search as it is composed of a topic mainly related to a professional interest (information visualization) and a topic related to a personal interest (Serge Gainsbourg). Then we asked each participant to specify the 2 topics categories that they considered either as interesting or uninteresting. 5 categories were available for the information visualization topic and 19 for Serge Gainsbourg. Finally a list of results was generated with 4 algorithm configurations: with the basic formula, with the categories consideration (personalized per participant), with randomness levels of 0.5 and 1. All these results were randomized in a single list. The participants evaluated them with the Discovery Hub application (including the explanatory features). An evaluator was present during the test to help them and to collect their impressions for further research. The participant sample was composed of 6 females and 10 males of 31 years old on average, mainly computer scientists.

[7] http://dbpedia.org/resource/Information_Visualization
[8] http://dbpedia.org/resource/Serge_Gainsbourg

Our experimentation aimed to evaluate the interest and the surprise of the users regarding the perspectives they can explore about topics. The framework notably proposes operations that constrain (the criteria specification) or free the (randomness injection) the spread over the data graph in order to increase the users' interest or surprise. We wanted to measure the influence of the algorithm variants on both this interest and surprise. For a precise evaluation we proposed the following definitions to the participants. A result is surprising if:

- You discovered an unknown resource or relation.
- You discovered something unexpected.

A result is interesting if:

- You think it is similar to the topic explored.
- You think you will remember or reuse it.

Users were invited to evaluate the interestingness and surprisingness of each result for each topic by indicating their degree of agreement or disagreement about the four following statements (presented in the form of a 4-point Likert scale):

- S1: *This result in itself is surprising: Not agree at all 1-2-3-4 Totally agree ;*
- S2: *This relation between the topic searched and the result is surprising: Not agree at all 1-2-3-4 Totally agree ;*
- S3: *This result is interesting: Not agree at all 1-2-3-4 Totally agree ;*
- S4: *This result is too distant from the topic searched: Very close 1-2-3-4 Too distant*.

8.4.3 Results

Hypothesis 8 and 9. The first interesting observation is that the selections of categories were very diverse among the users. Only 2 criteria selections on 16 appeared twice for the information visualization topic and only 1 for Serge Gainsbourg. It confirms that regarding a topic the users are interested in different aspects. Consequently allowing the exploration of topics through different perspectives might be useful to finely match the users' interest. Looking at the figure 8.7 we observe that the results generated by the algorithm using the criteria specification are judged more interesting than the results generated by the other algorithms thus the hypothesis 8 is validated. Conversely, we observe that these results are judged a bit less surprising thus the hypothesis 9 is not validated. Otherwise the loss in terms of surprise is minor and do not require in our sense a modification of the algorithm. The loss of surprise might be due to the prior knowledge of the users' about the criteria of interest they specified. Concerning the agreement the standard deviation was of 0.54 on average for all the different metrics and algorithm variants. The maximum average standard deviation was 0.68 (surprisingness of the relation, 0.5 randomized variant) and the minimum was 0.37 (perceived distance, basic formula).

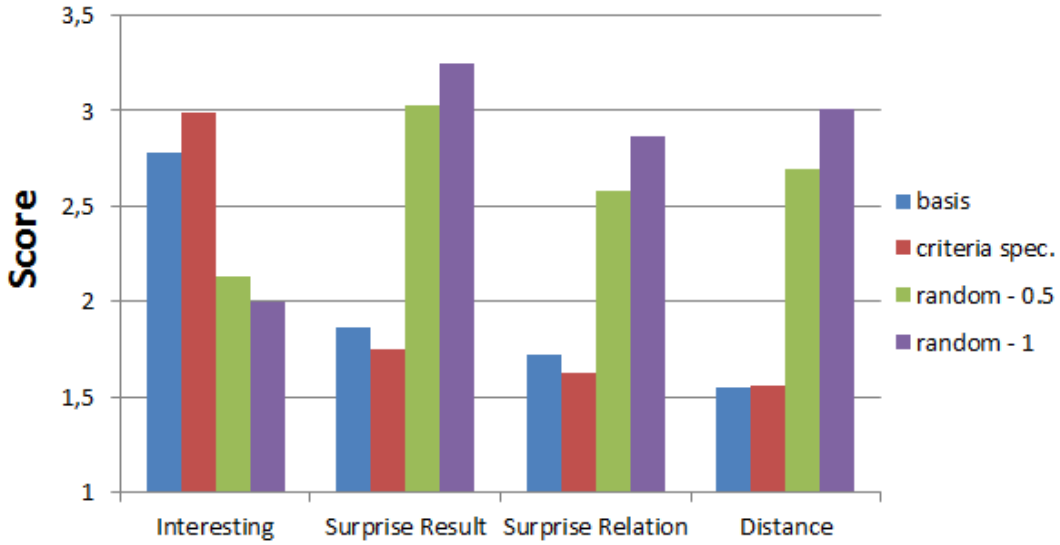

Figure 8.7: Interest, surprise and perceived distance of results according to 4 algorithm configuration

Hypothesis 10 and 11. We also observe that the results with a randomness set at 1 are judged more surprising than the ones with a randomness set at 0.5. Thus the hypothesis 10 is validated. We also observe that a majority of the results are judged irrelevant (>2.5). Thus the hypothesis 11 is not validated. It is also informative to observe the intersection between the relevance and the surprisingness. The intersection of the results evaluated both as very interesting and very surprising is also in favor of the 0.5 randomness value. Indeed, their percentage reaches only 3.3% for the randomness value of 1 versus 7.5% for the 0.5 value and approximately 4.5% for the other algorithms. Lower levels of randomness should be used to obtain a better trade-off between relevance and surprise.

8.5 Toward a complete evaluation of Discovery Hub

The experimentations previously described in this chapter were mainly focused on the users' perception about the results retrieved. These evaluations were essential as the approach we propose is largely based on algorithms. Otherwise these evaluations do not cover important aspects of the application such as the human-computer interactions, users' engagement, cognitive support or knowledge outcomes. We present below an ongoing reflection that aims to propose an evaluation covering all these aspects.

8.5.1 Protocol overview

The novel protocol we started to experiment aims to embrace the complexity in terms of human-computer interactions and knowledge acquisition that are characteristic of exploratory search[9]. It has 2 major objectives:

- **Verify that the search task executed is an exploratory one**. As we are unsure that our protocol will succeed to provoke an exploratory search behavior we check this aspect.
- **Verify that the system has positive effects over the exploratory search task execution**. For this we rely on the *systems desired effect* identified in chapter 2.

For this we introduce 3 innovative components in our protocol:

- The topic explored is chosen by each participant at the beginning of the search session.
- The participants' memory is used as a system efficiency metric.
- The users' intents are analyzed a-posteriori thanks to search session screencasts.

Personalized topic selection. Contrary to the previous evaluations the explored topic is chosen by the participants at the beginning of the evaluation-session. Relying on users' interest rather than on an assigned topic aims to increase their engagement. Consequently there is no task or scenario assigned to them. They use the tool independently from any assignment, in a *natural* way. They are free to decide what is their initial search objective and to make it evolve without any artificial constraint.

At the beginning of the evaluation the evaluator asks the following question: *do you have any passion or interest about which you want to discover new information?*. We identify their initial search objective from their answer e.g. discovering the topic, gaining more knowledge, being surprised about it. Then the participants are asked to evaluate their knowledge about the chosen topic thanks to a 5-points Likert scale. Before the evaluation search-session begins a topic is chosen in the Discovery Hub random topics suggestion page in order to demonstrate the application.

Users' memory as an efficiency metric. During the evaluations of the advanced querying functionalities we specified that a result is interesting if *the user thinks he will remember it or reuse it*. In this protocol we explicitly consider the users' memory as a metric of the system efficiency. For this we use several free-association tests. In the free-association test *"the subject is told to state the first word that comes to mind in response to a stated word, concept, or other stimulus*[10]. It is a simple technique for estimating the users' state-of-knowledge about a topic. In our protocol 1-minut free-association tests are performed before and right after the search

[9]The protocol is designed with Alain Giboin and Émilie Palagi, the first experimentations and consecutive result extraction was performed by Émilie Palagi

[10]http://global.britannica.com/EBchecked/topic/647931/word-association-test"

session and then once again 1 week after. The duration of 1 week has been chosen to ensure that the knowledge is stored in the long-term memory of the users. The differences in the free word association lists (referred to as FWA below) are quantified and constitute an indicator of how many new important results have been discovered by the users. Several questions are also asked at the end of search session to get qualitative feedback about the knowledge gained. The first question is *have you learned new information about the topic* and to list these information. The second one is *were you surprised by some new information* and to list them again.

Screencast-based users' intent analysis. The search session per participant has a duration of 20 minutes. It is fully recorded with the Silverback application[11]. Right after the session the users are asked to comment the video by especially focusing on the intent behind each the interactions they made. The 2 videos were synchronized and ELAN[12] was used for annotating them in order to ease their analysis. The screencasts were also used to analyze the semantic field employed by the users when they describe the intent behind their actions e.g. *understand*, *discover*, *deepen*. The videos also served to measure several users' behavior metrics. Such metrics are presented in the next subsection, they include the number of queries processed, the amount of results consulted, etc.

8.5.2 Minimizing the evaluation difficulties

Let us now first list the difficulties inherent to the evaluation of the exploratory search systems and then detail how the protocol we design should minimize them.

- **The traditional information metrics (recall, precision, completion time) are unadapted**: we employ different measures that show the searcher behavior, intents and memory.
- **The elaboration of exploratory search task scenario is difficult**: we do not use a task-based approach, the motivation of the participants is endogenous and personal as it resides in their interest of the chosen topic.
- **There is a high-level of subjectivity**: the fact the participants choose their exploration topic minimizes such subjectivity. Indeed, assigning a topic to the users exposed us to subjectivity biases during the previous evaluations. During the first experimentation the participants have not always seen the seed-movies. During the third one the singer Serge Gainsbourg was totally unknown from some participants.
- **The users' fatigue limits the evaluation possibilities**: only 20 minuts is dedicated to search, which is cognitively intensive. The users intents are analyzed in the second time and the long-term memory effect is tested one week later. The evaluation is distributed over the time, minimizing the effect of the fatigue.

[11] http://silverbackapp.com/
[12] https://tla.mpi.nl/tools/tla-tools/elan/

- **The very diverse designs of the exploratory search engines make them hardly comparable**: the protocol has been conceived to be totally system-independent by focusing the evaluation on the effects of the systems.

8.5.3 Hypothesis and metrics

We list the hypothesis we want to verify below. We make the correspondence with information and metrics obtained thanks to the protocol for each of them. The first set of hypotheses corresponds to the objective of verifying that the search task executed is an exploratory one. They are extracted from the list of exploratory search task characteristics summarized in chapter 2, see page 13.

- **Hypothesis T1: The goal is to learn or investigate**: we check that the verb they use to qualify the participants initial search objective mentions a learning or investigating purpose.
- **Hypothesis T2: The search task is open-ended**: we check if the memory-features of the system are used. We also observe the amount of entries present in the third free-word association (FWA3) tests that were not present in the first one (FWA1). They represent potential new exploration axes.
- **Hypothesis T3: The information need is evolving**: we observe the amount of orienteering actions perfomed during the search session (queries in Discovery Hub).

The second subset of hypotheses presented hereafter aims to verify that the system has positive effects over the exploratory search task execution. The hypothesis are based on the desired effects of exploratory search systems identified in chapter 2, see page 13.

- **Hypothesis E1: The users explore multiple results and browsing paths:** number of results opened and average length of the browsing paths.
- **Hypothesis E2: The system inspires the users and shapes the information need:** Number of orienteering actions starting from a result (result pop-up *run an exploration* button in Discovery Hub). Number of new entries in FWA2 that reflect a change in the users' topic mental representation.
- **Hypothesis E3: The system favors discoveries:** we rely on the answer to the question *were you surprised by some new information*.
- **Hypothesis E4: The system eases memorization:** for this we observe the amount of new entries in FWA3. We also rely on the answers to the question *have you learned new information about the topic*.

8.5.4 Preliminary results

The protocol has been designed to be largely system independent. Its value resides in the comparison of several exploratory search systems on a fair basis.

The protocol will be probably re-adjusted when the first system comparison will be done. To have an idea of what can be the outcomes of this experimentation and if the protocol was feasible we tested it on 3 participants, see figure 8.8. The screencasts of their seach-session was put online[13]. The needed metrics were extracted and the interactions sequences were modeled[14], see appendix I on page 253. We present the information we get from these 3 exploratory search sessions below:

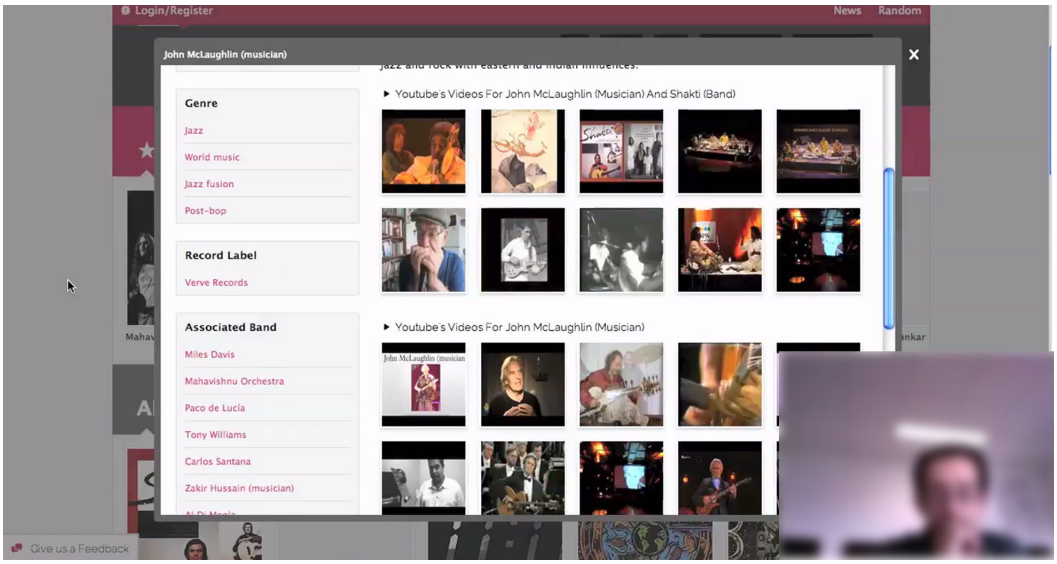

Figure 8.8: A participant (on bottom-right) commenting the screencast of his exploratory search session

Objective 1: exploratory search task cheking.

- **Hypothesis T1: The goal is to learn or investigate**: initial search objectives mentioned by the users were: user1 (U1)*"learn new information about the history of free-jazz"*; U2 *"discover new board games"*; U3 *"learn new information about the Senegal"*.
- **Hypothesis T2: The search task is open-ended**: 1 out of 3 participants used the memory features. We observed new entries in all the FWA2 (8/15, 5/8, 8/13) and new entries for 2 participants in the FWA3 (2/17, 2/7, 0/9).
- **Hypothesis T3: The information need is evolving**: the users entered a variable amount of queries. Some of them were executed using the search bar: U1: 7; U2: 5; U3: 3. The other queries were launched from a result pop-up using the *"run an exploration button"*: U1: 16; U2: 5; U3: 0.

[13]Playlist *Expe - search session screencasts*: https://www.youtube.com/playlist?list=PLBz-BpzPSRtZYuxqxYDxAk4JpQa2QyMAl

[14]By Émilie Palagi

Objective 2: desired effects of the system measurement.

- **Hypothesis E1: The users explore multiple results and browsing paths:** The amount of results pop-up visited by each user was also very variable: U1: 47; U2: 23; U3: 19. The average length of the browsing paths inside the results pop-ups was the following one: U1: 7.6, U2: 2.75, U3: 1.
- **Hypothesis E2: The system inspires the users and shape the information need:** The *run an exploration* button was more or less frequently used as a search pivot depending on the participant: U1: 16; U2: 5; U3: 0. The new entries present in FWA2 also reflect the fact that the users were inspired by their search session: U1: 8 new entries on a total of 15 words; U2: 5 out of 8; U3: 8 out of 13.
- **Hypothesis E3: The system favors discoveries:** the users all answered *yes* to the question *were you surprised by some new information*.
- **Hypothesis E4: The system eases memorization:** they the users answered *yes* to the question *have you learned new information about the topic?*. We also observed that new entries appeared in FWA3 (when compared to FWA1) i.e. new words are associated to the topic in the users mental model: U1: 2 new entries out of 17; U2: 2 out of 7; U3: 0 out of 9.

The protocol needs now to be executed on a larger group of participants and to be compared to the results obtained with other systems

8.6 Query-log statistics

Finally, since DH is online and used by several users, we now have a new source to analyze the systems: its query log (approximately 2.400 queries at the time of writing). The figure H.2 shows the percentage of queries per class which inform us about the domains of interest queried in Discovery Hub. The first observation is that approximately half of the queries corresponds to untyped resources (*Miscellaneous* on the figure). It confirms our choice to not exclude the untyped resource from the computation. Second we observe a strong interest about the culture domain and more particularly the music, cinema and literature ones. We can observe the same tendency in the table 8.2 which shows the occurrences of the categories associated to the queries. Several rare categories of cultural items are very popular e.g. *Ivor_Novello_Award_winners*. One explanation of this good representation of cultural resources is that Discovery Hub might be used mainly as a discovery engine for such items where there is a strong interest for recommendation. However it is observable that there is a high variety in the topic that have been queries. There is an important long-tail concerning a wide variety of topic-classes that have been queries only once e.g *SpaceMission, Fish, BasketballPlayer*. This high variety of

queries is also visible on the random suggestion of topics[15] available on Discovery Hub, see figure 8.10. This confirms the interest to conceive a cross-domain exploratory search system that might not only cover a wide variety of information needs but also create unexpected bridges between the users' interest. Moreover, several domains retrieve very complete results such as the medical anatomy one e.g. queries related to bones, veins, organs. Numerous use-cases well-supported by Discovery Hub are still to be discovered.

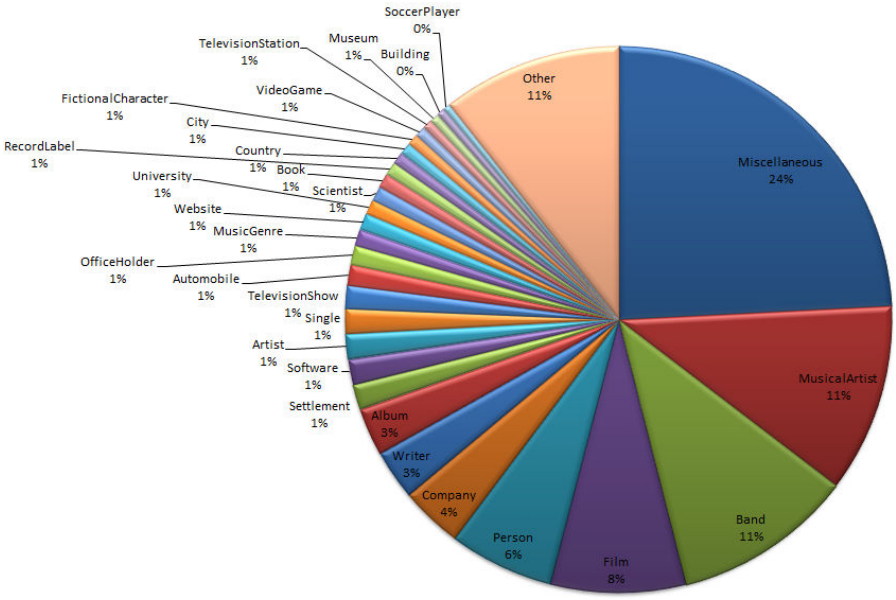

Figure 8.9: Percentage of queries by class in the Discovery Hub query-log

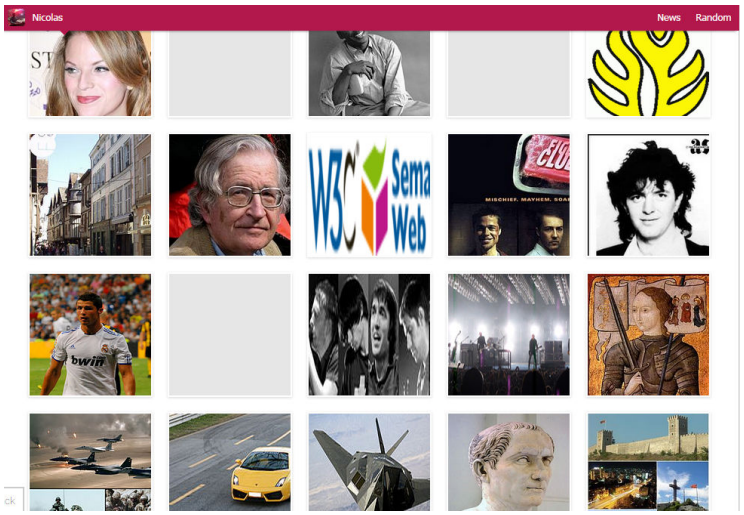

Figure 8.10: Suggestion of random topics to search on the Discovery Hub interface

[15] http://discoveryhub.co/random

	Category	Query-log hits	Category instances
1	Living_people	496.0	1046812
2	Grammy_Award_winners	129.0	4202
3	English-language_films	122.0	45996
4	American_films	54.0	30833
5	BRIT_Award_winners	53.0	470
6	American_film_actors	52.0	28202
7	Musical_quartets	51.0	1957
8	Article_Feedback_Pilot	43.0	8304
9	Rock_and_Roll_Hall_of_Fame	38.0	450
10	American_television_actors	36.0	23554
11	American_male_singers	35.0	6357
12	People_from_Paris	34.0	4804
13	English-language_singers	33.0	1484
14	American_rock_singers	32.0	1787
15	Virgin_Records_artists	27.0	306
16	English_rock_music_groups	27.0	2394
17	Musical_quintets	26.0	1721
18	American_singer-songwriters	25.0	3641
19	Ivor_Novello_Award_winners	24.0	492
20	American_Jews	24.0	7824

Table 8.2: Occurrences of the categories associated to the queries from the Discovery query-log

8.7 Conclusion

In this chapter we reminded the difficulty of evaluation inherent to exploratory search systems. There is a need to pursue the research about the evaluation protocols in order to better compare the systems. We presented the evaluations we made for the monocentric, polycentric algorithms and for the advanced querying variants. Our approach is strongly based on an algorithm that automatically selects and ranks results of interest to explore. Thus we needed to evaluate the perception of the retrieved results. We only relied on the human judgment due to the nature of exploratory search, that implies a high users involvement and a lot of subjectivity. We compared ourselves to a baseline for the mono-centric queries but we were not able to find a valuable comparison basis for the polycentric and the advanced querying variants. In the last case we compared several algorithm configurations amongst themselves and observed in particular the trade-off between the results relevance and their surprisingness. Overall the three experimentations gave positive results. However it also showed that slight algorithmic adjustments were needed e.g. lowering the randomization threshold.

These evaluations constitute a solid basis but does not cover yet all the important exploratory search needs covered by Discovery Hub. The only components

of the interface that were studied at the time of writing, are the explanatory features. They are very important in Discovery Hub as they help the understanding of non-trivial results. They received very positive feedbacks from the users. We need now to evaluate the complete interaction model and the interface of the application by observing if it produces the desired effects identified in chapter 2. We started to design a protocol aiming to overtake our previous experimentations. This protocol introduces several experimentation innovations. It will be improved and fully executed outside the scope of this thesis. In order to give a view on this ongoing reflection we presented in this chapter the first results we obtained with 3 participants.

At the end of the chapter the composition of the Discovery Hub query-log was mentioned. We can observe a predominance of queries related to cultural domains. Otherwise it is noticeable that there is a high variety in term of topics explored. This tends to confirm that it is relevant to conceive exploratory search algorithms and systems that are not constrained to a specific domain.

CHAPTER 9

Conclusion and perspectives

In the following conclusion we summarize the context of our research. In a second time we recall our contributions that aimed to answer the principal research questions asked in the introduction. Last but not least we open the reflection by listing the opportunities we see for linked data based exploratory search systems. We reveal the short-terms improvements we see for Discovery Hub as well as the new research questions that emerged along this thesis.

9.1 Research summary

Before summarizing the research context and the contributions of this thesis we remind the 4 research questions that guided the author:

- (i) how can we discover and rank linked resources to be explored starting from the user topic(s) of interest?
- (ii) how to address remote linked data source for this selection?
- (iii) how to optimize such data based exploration approach at the interaction level?
- (iv) how to evaluate such exploratory search systems?

9.1.1 Research context

The general topic of the thesis is search. It focused more specifically on how to leverage the data semantics richness for successful exploratory search.

The chapter 2 set the usage context of the thesis. Even if search engines are very popular today the search approaches and technologies still have an important room for improvement regarding complex queries and diverse information needs. In this context we specifically defined and positioned exploratory search (subsection 2.3.1, page 11). We also detailed why the conception (subsection 2.3.3, page 17), comparison and evaluation of such systems are difficult (subsection 2.3.4, page 23). An important contribution of this chapter is the identification of the desired effects of exploratory search systems starting from the characterization of the search tasks they aim to support (paragraph 2.3.3, page 19). These desired effects are used all along the thesis to structure the reflection.

The chapter 3 set the technological context of the thesis. It gives an overview of semantic search which refers to the practice of enhancing search technologies

by the processing of semantics. The chapter starts from a data point of view by insisting on the massive proliferation of structured data on the web today (section 3.2, page 28). It focuses more particularly on the linked open data cloud and on its central dataset: DBpedia (section 3.4, page 33). The LOD is the largest source of structured data and can be considered as the main achievement of the semantic web vision at the time of writing. The chapter continues by outlining the main components of the semantic search approaches (section 3.5, page 43). It is a very active research topic and the major search engines are increasingly using semantics to enhance their services.

The chapter 4 is a prolongation of the chapter 2 and 3 is dedicated to exploration and discovery using semantic data. It details a state-of-the-art review of such approaches and systems within broad areas of classification. The semantic browser (section 4.2, page 59), recommenders (section 4.3, page 79) and exploratory search systems (section 4.4, page 87) are detailed. An important contribution of this chapter is the synthesis of the research advancements in its discussion part (section 4.5, page 94). The evolution of the research over the time is discussed and the most advanced systems are compared in a matrix. The matrix focuses on their main information retrieval and human-computer interactions aspects. Finally the limits, achievements and opportunities found in the literature are identified for each of these aspects. The identification of these opportunities constitutes the pivot toward the contribution part of the thesis. A compact version of this state-of-the-art review was published in [125].

9.1.2 Contributions

The exploratory search systems are human-computer interaction retrieval systems. Their information retrieval components are as important as the human-computer interaction ones. The objective of the thesis was to offer a complete exploratory search system: from the formalization of the algorithms to the interaction model and interface. The evaluation of the whole also constituted an important part of the thesis work.

The chapter 5 answers the research question (i). We decided to ground our approach on algorithms that automatically selects and ranks informative result sets to explore. It allows to efficiently support an orienteering behavior at the beginning of exploratory search, when the users lack knowledge and need support to evolve in the information space. We started from a spreading activation basis (section 5.2, page 106) and proposed an adaptation optimized for typed graph. The specificity of our algorithm is that it leverages the data semantics to drive the propagation in relevant zones of the graph only and favor the activation of nodes that are semantically similar to the query-node(s). A strong advantage brought by spreading activation is the flexibility it gives in term of querying. Indeed we first formalized the simplest queries possible i.e. the monocentric ones (section 5.4, page 115), published in [128]. Starting from this formalization we proposed later the formalizations of several advanced querying modes. It included composite querying thanks

to polycentric spreading activation (section 5.5, page 122), published in [123], as well as the criteria of interest specification and controlled randomness injection variants (section 5.6, page 127), published in [126]. The design of the algorithms is discussed at the end of the chapter.

The chapter 6 answers the second research question (ii). It explains and motivates the two paradigmatic design choices we made (section 6.2, page 137). First the results have to be computed at query-time. Second the data are consumed remotely from distant SPARQL endpoints. Our approach consists in incrementally importing at query-time a sub-graph from the targeted SPARQL endpoint using SPARQL query with the service operator. The neighbors of the most activated nodes are imported till a limit is reached. The algorithm is applied locally on the subgraph imported. This method allows us to reach the required level of flexibility in terms of data selection and querying expressiveness. Nevertheless it also introduces a triangular trade-off between the size of the import, the response-time and the results relevance. The chapter 6 discusses the feasibility of the method without taking the results relevance in consideration (this is the purpose of the chapter 8). It presents in details the extensive analyses we made to observe the monocentric (section 6.3, page 143) and polycentric (section 6.4, page 152) algorithms behaviors over the main dataset used in this thesis: DBpedia. These analyses helped to set several important calibration parameters such as the maximum number of iterations and the limit of triples locally imported. They were respectively published in [128] and [123]. Finally this chapter discussed the applicability of the algorithm outside the DBpedia context (section 6.6, page 167). For this we mainly relied on analyses over randomly generated graphs. The principal findings of these analyses are that the algorithm quickly converges, that its convergence is correlated with the diameter of the targeted dataset.

The chapter 7 answers the research question (iii) by proposing a novel interaction model and web application (section 7.2, page 176). The application is online; it was showcased during several important semantic web conferences and various technological events. It won the ESWC2013 best demonstration award [127] and was presented at the ISWC2014 conference with a focus on the advanced querying modes [124]. The chapter 7 notably explains the Discovery Hub design choices of the actual version by comparing it to the interface of the first version (subsection 7.3.1, page 188). It also discusses the interaction model regarding the guidelines available in the literature (subsection 7.3.2, page 197) and the desired effects of the exploratory search systems previously identified in chapter 2 (subsection 7.3.3, page 198).

The chapter 8 aims to validate the contributions proposed in the previous chapters by relying on several users' experimentations. We deliberately performed only users' evaluations as the human engagement during exploratory is critical. As our approach strongly relies on automatic resources selection and ranking the evaluations were mainly focused on the algorithm results relevance and unexpectedness for the monocentric (section 8.2, page 204), polycentric (section 8.3, page 209) and advanced querying variants (section 8.4, page 213). These evaluations were respec-

tively published in [128], [123] and [126]. The chapter also addresses the research question (iv) by proposing a novel protocol that integrates innovative experimentation approaches (subsection 8.5, page 216). This protocol aims to evaluate the Discovery Hub interaction model and to overtake the common experimentations limits cited in the literature. More precisely the innovations concern the personalized selection of the topic explored, the total absence of assignment to motivate the exploration (e.g. task-based scenario or objectives), the users' memory being used as an efficiency metric and the use of intent-analysis using search sessions screencasts. At the time of writing a first round of experimentation with 3 users was executed.

9.1.3 Publications

To sum up this thesis led to the following main contributions:

- A state-of-the-art review of the exploration and discovery approaches based on linked data as well as its synthesis and analysis.
- The formalization of a core algorithm and its variants for automatically selecting and ranking a set of results to explore starting from the users' topic(s) of interest.
- An innovative implementation method of such algorithms that computes the results at query-time from distant data. This approach allows to reach a high level of flexibility in terms of data selection and querying.
- The understanding and calibration of the algorithm behavior inside and outside the DBpedia context.
- The Discovery Hub web application implementing the framework and proposing an interaction model optimized for exploratory search.
- The evaluation of the users' perception about the results relevance and unexpectedness for the monocentric, polycentric queries as well as for the criteria of interest specification and the controlled randomness injection variants.
- The evaluation of the efficiency and the users' perception about the explanation features in the context of monocentric and polycentric queries.
- The proposition of an innovative evaluation protocol for exploratory search systems.

These contributions were published in several conferences having a partial or complete focus on semantic web research. Discovery Hub notably won the best demonstration award at ESWC2013. The publications related to Discovery Hub and to the field of linked data based exploratory search are listed here-after:

- Nicolas Marie, Fabien Gandon. **Demonstration of multi-perspective exploratory search with the Discovery Hub web application**, *ISWC2014, Riva Del Garda, Italy* (demonstration)

- Nicolas Marie, Fabien Gandon.**Survey of linked data based exploration systems,** *IESD2014, Riva Del Garda, Italy* (long paper)
- Nicolas Marie, Fabien Gandon, Alain Giboin, Emilie Palagi. **Exploratory search on topics through different perspectives with DBpedia**, *Semantics 2014, Leipzig, Germany* (long paper)
- Nicolas Marie, Fabien Gandon, Myriam Ribière, Florentin Rodio. **Discovery Hub: on-the-fly linked data exploratory search**, *I-Semantics 2013, TU Graz, Austria* (long paper)
- Nicolas Marie, Fabien Gandon, Damien Legrand, Myriam Ribière. **Exploratory search on the top of DBpedia chapters with the Discovery Hub application**, *ESWC2013, TU Graz, Austria* (best demonstration award and poster)
- Nicolas Marie, Olivier Corby, Fabien Gandon, Myriam Ribière. **Composite interests' exploration thanks to on-the-fly linked data spreading activation**, *Hypertext 2013, Paris* (long paper)

Research material was also published to be reused by the community. It includes several pieces of code, the DBpedia sampler and sample, the screencast videos. Discovery Hub also served as an academic research support, some of the corresponding reports are available online[1] [2]. During this thesis the author was involved in 7 other publications as main author or co-author. These publications concerned social networking and web sciences topics[3]. Although they are not linked to the thesis directly we list them here:

- Clare J. Hooper, Nicolas Marie, Evangelos Kalampokis, **Dissecting the Butterfly: Representation of Disciplines Publishing at the Web Science Conference Series**, *Web Science 2012, Northeastern university, Evanston, United States, 2012* (short paper).
- Nicolas Marie, Fabien Gandon. **Advanced social objects recommendation in multidimensional social networks**. *Social Object Workshop 2011, MIT, Boston, USA* (long paper).
- Nicolas Marie, Fabien Gandon, Myriam Ribière. **Pervasive sociality : advanced social objects recommendation**. *Web Science 2011, Koblenz, Germany* (poster).
- Nicolas Marie, Fabien Gandon, Myriam Ribière. **L'ontologie OCSO : une ontologie pour le futur du web social**. *IC2011, Chambéry, France* (poster).
- Johann Stan, Myriam Ribière, Jérôme Picault, Lionel Natarianni, Nicolas Marie. **Semantic-Awareness for a Useful Digital Life**. *IGI Global Book: Social Network Analysis and Mining, 2010.* (book chapter).

[1] http://issuu.com/juneviendalmare/docs/designthinkingtoolset4developers

[2] http://atelierihm.unice.fr/enseignements/wp-content/uploads/sites/3/2013/12/CEIHM-Gr5-Discovery-Hub-Rapport-interm%C3%A9diaire.pdf

[3] http://ncmarie.tumblr.com/publications

9.2 Perspectives

All along the thesis new ideas of exploratory search functionalities arose. The inspiration was catalyzed by the DBpedia richness. This dataset offers plenty of possibilities to enhance the data processing, create new algorithms variants and support novel interactions. The fact that we had a functioning web application also brought a lot of ideas. However the exploratory search systems designers need to pay attention to maintain the coherence and intuitiveness of the interaction model and interface. Indeed, these inspirations can lead to a profusion of functionalities that can result in an interface that is confusing to the users. We present the perspectives in two subsections below. The first one is dedicated to the short-term improvements that do not require extensive research. The second one presents the long-term improvements that open difficult research questions and require extensive research.

9.2.1 Short-term improvements

The potential improvements presented below can be considered as "*incremental innovations*". They were not implemented due to lack of time and development resources. They do not fundamentally change the Discovery Hub application functioning so they don't require an extensive research effort, but can be expensive in term of development:

- **Facet and collection-levels interactions**: both automatic (results list facets) and user-driven (collection) organization of results exist in Discovery Hub. It would be very useful to propose resources set-level of interactions (the set being a collection or a facet). It can includes for instance visualizing all the geo-tagged resources on an interactive map, which propose other services at its turn. It would be even more interesting to trigger third-party service interactions from Discovery Hub e.g. generating a music-streaming service playlist from a *Band* facet of a result list. Along with this idea, being able to compute summaries and overviews of sets of resources can also be helpful.
- **Collaborative exploratory search**: collaborative exploratory search is especially promising as it can gather the efforts of several users having different points of view, prior knowledge and search tactics. Discovery Hub actually implements a variety of social functionalities but lacks mechanisms to foster the collaboration amongst its users. It would be valuable to implement public or group-restricted collections as well as to recommend people sharing common interests. Otherwise turning Discovery Hub into a collaborative exploratory search engine requires also to implement collaboration functionalities such as messaging and interaction tracking. It represents a consequent amount of development. An option is to implement it by integrating Discovery Hub in an existing platform supporting collaboration.

- **Post-query re-ranking mechanisms**: an interesting improvement for Discovery Hub would be to offer re-ranking, re-sorting functionalities after the query is processed. Instead of launching several successive queries with different criteria of interest specified a user might be able to dynamically change the values associated to the criteria in order to re-rank the results list dynamically. In order to assist the users in these *re-ranking* operations it would be valuable to make explicit the influence of the criteria on the ranking of each result. In the same spirit the results of the composite queries could be rearranged by modifying the weight associated to the seeds composing the query. In the current implementation all the seeds have the same importance on a computing point of view.
- **New query-means**: Discovery Hub can benefit from going beyond the classical search bar by generating queries from other inputs than entities. The idea is to consider the application as a *hub* not only when considering the results but also the inputs. Queries can be triggered from the users' bookmarks, social networks publications, entities identified in the browsed web-pages, or from the closest geo-tagged resources (using *geo:long* and *geo:lat* properties present in DBpedia for instance). Third-party services such as Shazam[4] or Google Goggles[5] can be used to support multimedia inputs. Ultimately we can see Discovery Hub as a platform that integrates the users' interests gathered from multiple services and allowing them to discover unexpected relations from the whole. Of course such integration raises important privacy concerns.

9.2.2 Long-term improvements

In this part we present the research directions that are interesting to conceive the next generation of linked data based exploratory search system. They open several research questions that necessitate important research efforts.

- **Negative search:** we can imagine the possibility of performing negative spreading activations in a linked dataset in order to identify weak signals in the remaining resources. Combining positive and negative queries can support new use-cases, e.g. it can be a way to eliminate the influence of strongly connected nodes to observe the remaining connections. Otherwise it poses numerous questions on the algorithm behavior at a polycentric level. The question of how to explain this querying mode to the users is also a hard point.
- **Adaptive queries recommendation:** it would be valuable to propose a system of query recommendations in order to help the users to explore a topic from multiple relevant angles. Such queries can be proposed by identifying categories of interest in the users' collections that match the currently

[4] www.shazam.com
[5] https://support.google.com/websearch/answer/166331?hl=en

retrieved result set. The level of randomness can also be increased as the users gain knowledge on a topic. However all the query recommendations should be explicated and understood by the users. It requires the use of a sophisticated user profile representation as well as personalization methods.

- **Collection-based knowledge creation:** it seems particularly promising to turn the collection into active information assets. The collections represent, to a certain extent, the state-of-knowledge of the users about a topic. We can imagine to perform collection-based recommendations by launching composite queries with all the contained resources in order to identify new potential resources of interest. Another interesting perspective is to identify the knowledge at the cross-road of several collections.
- **On-the-fly distributed data selection:** a difficult but promising perspective is the automatic identification and merging of the best LOD dataset regarding a query. Numerous criteria can be taken into account to select them such as the data freshness, the specialization of the linked dataset, the amount of triples related to the topic of interest, etc. It is in line with the semantic vision where the distribution is a key concept. Moreover the software architecture proposed by Discovery Hub can be an interesting starting point for an implementation. However the query-dependent ranking of the data sources, the performances, the algorithms behavior and the presentation of multi-source data in a unique interface all represent difficult research problems.
- **Massive use and commercial success:** it is important that exploratory search systems become popular, appreciated and massively-used. They can bring significant improvements to important cognitive activities such as learning and decision-making. On the users' side this notably involves the creation of a new space and audience on the web for alternative search methods. On the systems side it involves the scalability of the approaches in terms of data storing and processing. At the time of writing we are at an interesting moment of the history of linked data based exploratory search systems. Some of the searchers cited in this thesis created their start-ups[6] [7]. At the same time the three major search players released their knowledge graphs and panels. We forecast a tough competition from which successful solutions will emerge and benefit to the users.

[6] https://developer.seevl.fm/
[7] http://www.sepage.com/

APPENDIX A

DBpedia 3.6 network metrics

Network info

Code	DB
Category	Misc
Data source	http://wiki.dbpedia.org/Downloads
Vertex type	Entity
Edge type	Relationship
Format	Directed
Edge weights	Multiple unweighted
Size	3,966,924 vertices (entities)
Volume	13,820,853 edges (relationships)
Average degree (overall)	6.9680 edges / vertex
Fill	8.1723×10^{-7} edges / vertex^2
Maximum degree	472,799 edges
Reciprocity	3.88%
Size of LCC	3,915,921 vertices
Size of LSCC	178,593 vertices
Wedge count	174,275,329,593
Claw count	$1.9647361248950896 \times 10^{16}$
Triangle count	8,329,548
Square count	31,572,553,137
Power law exponent (estimated) with d_{min}	2.0710 ($d_{min} = 59$)
Gini coefficient	66.2%
Relative edge distribution entropy	87.4%
Clustering coefficient	0.0143%
Diameter	67 edges
90-percentile effective diameter	6.27 edges
Median shortest path length	5 edges
Mean shortest path length	5.19 edges
Spectral norm	731.84

Figure A.1: DBpedia 3.6 metrics according to the Koblenz network collection website

APPENDIX B

Kendall-Tau

τ_b is a rank correlation measure reflecting the concordance of two ranked lists where:

$$\tau_b = \frac{\sum_{i<j} sgn(x_i - x_j) sgn(y_i - y_j))}{\sqrt{(T_0 - T_1)(T_0 - T_2)}}$$

where:

$$T_0 = \frac{n(n-1)}{2}$$
$$T_1 = \sum_k \frac{t_k(t_k-1)}{2}$$
$$T_2 = \sum_l \frac{u_l(u_l-1)}{2}$$

and the t_k is the number of tied x value in the kth group of tied x values, u_l is the number of tied y values in the lth group of tied y values, n is the number of observations and $sgn(z)$:

$$sgn(z) = \begin{cases} 1 \, if \, z > 0 \\ 0 \, if \, z = 0 \\ -1 i \, f z < 0 \end{cases}$$

τ_b is comprised between -1 and 1: -1 means a total discordance and 1 a total concordance. In this thesis we use to it we observe the similarity of the rankings from iteration to another. It notably allows observing the algorithm convergence.

APPENDIX C

Visualizations

The colors of the arcs on the figure C.1 correspond to the level of activation that pass through them: white equals to unactivated, the activated arcs are colored from yellow (minimum value) to red (maximum). The correspondence between the colors and the activation values was manually set.

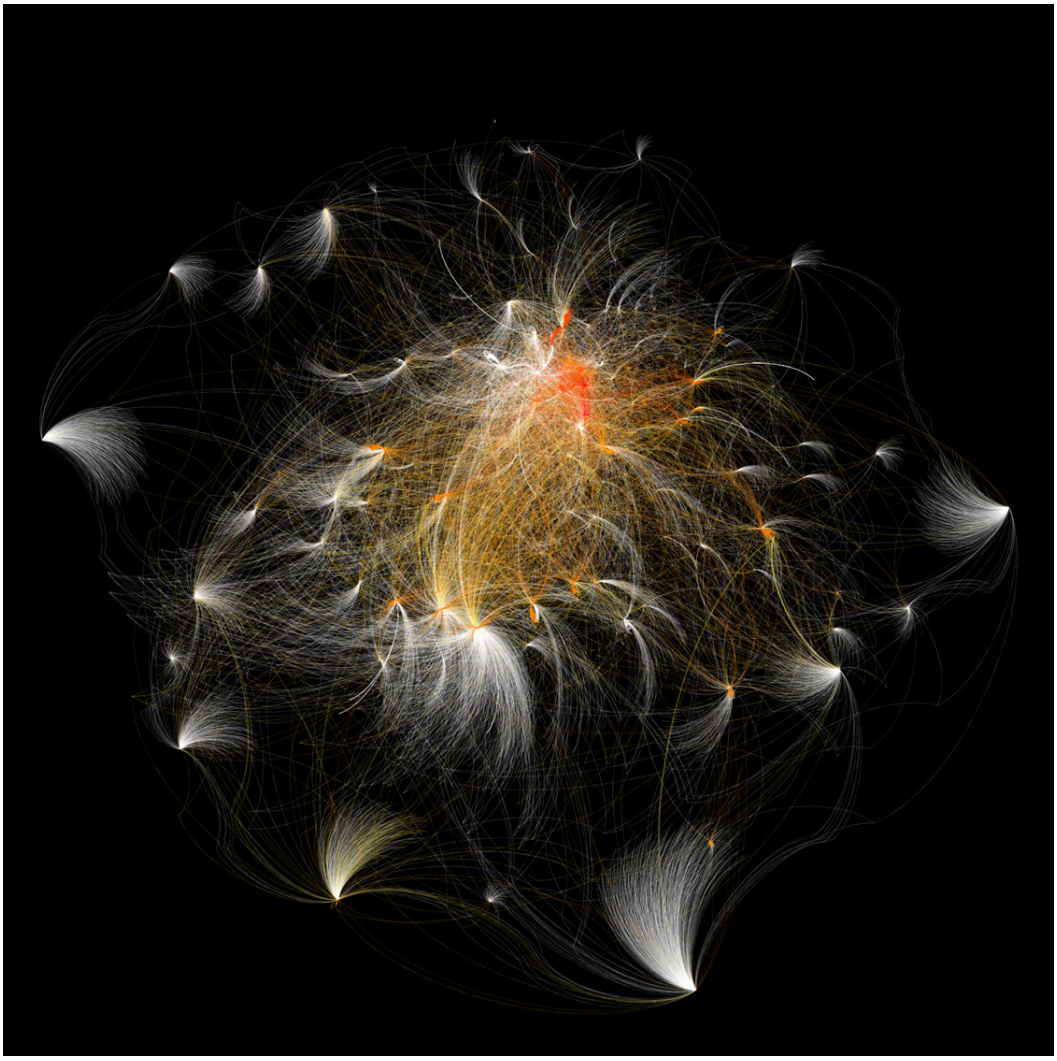

Figure C.1: Visualization of the activations at the last iteration in the 2-hops neighborhood, query Claude Monet, basis algorithm.

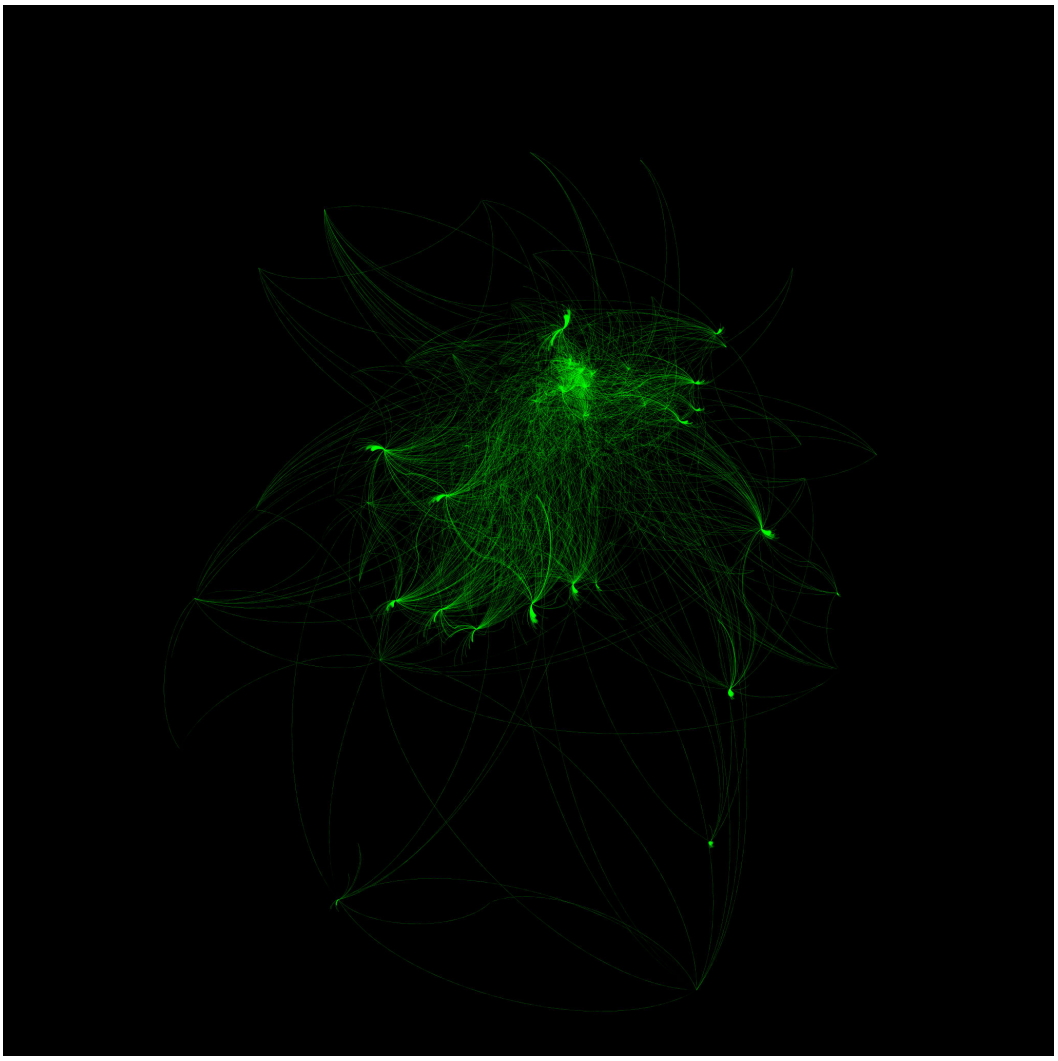

Figure C.2: Visualization of the top 100 results and their relations in the 2-hops neighborhood, query Claude Monet, criteria *"French, not impressionis"*

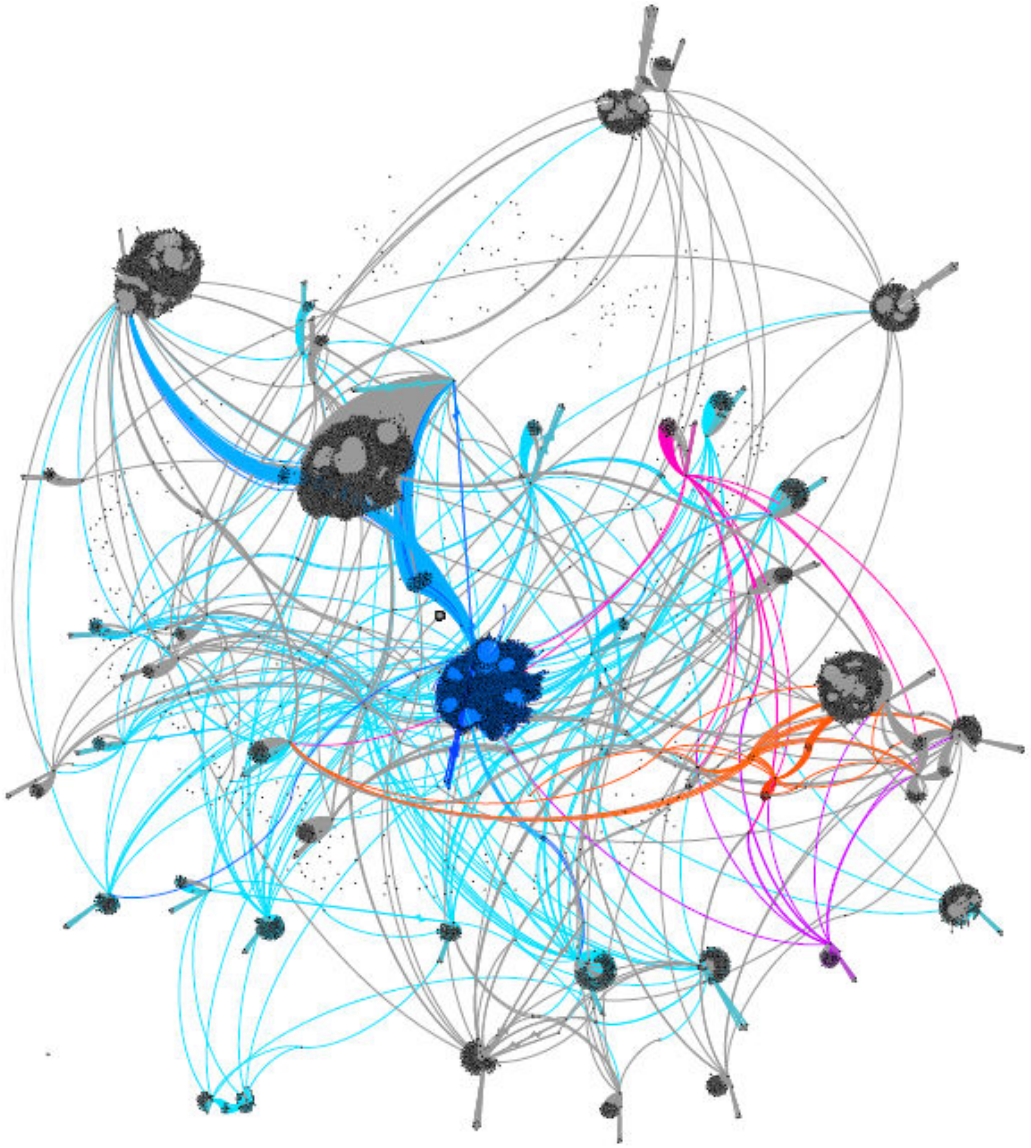

Figure C.3: Visualization of the third iteration of a polycentric query, Charles Baudelaire propagation in blue, Arthur Rimbaud in red and the nodes activated at a polycentric level in purple

Appendix D

Experimentations using the Discovery Hub interface

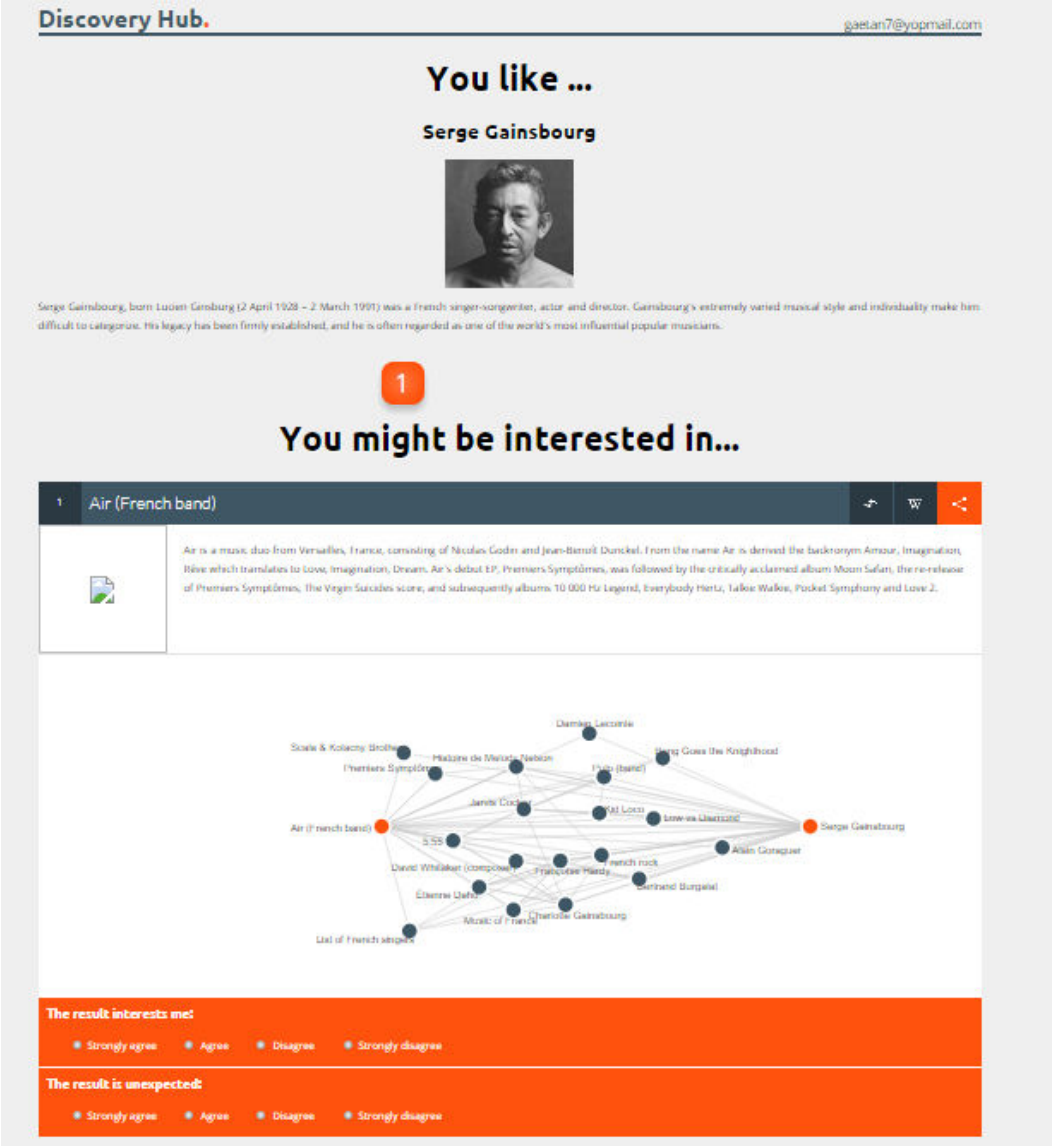

Figure D.1: Experimentation interface using the Discovery Hub application

APPENDIX E

Functional modeling

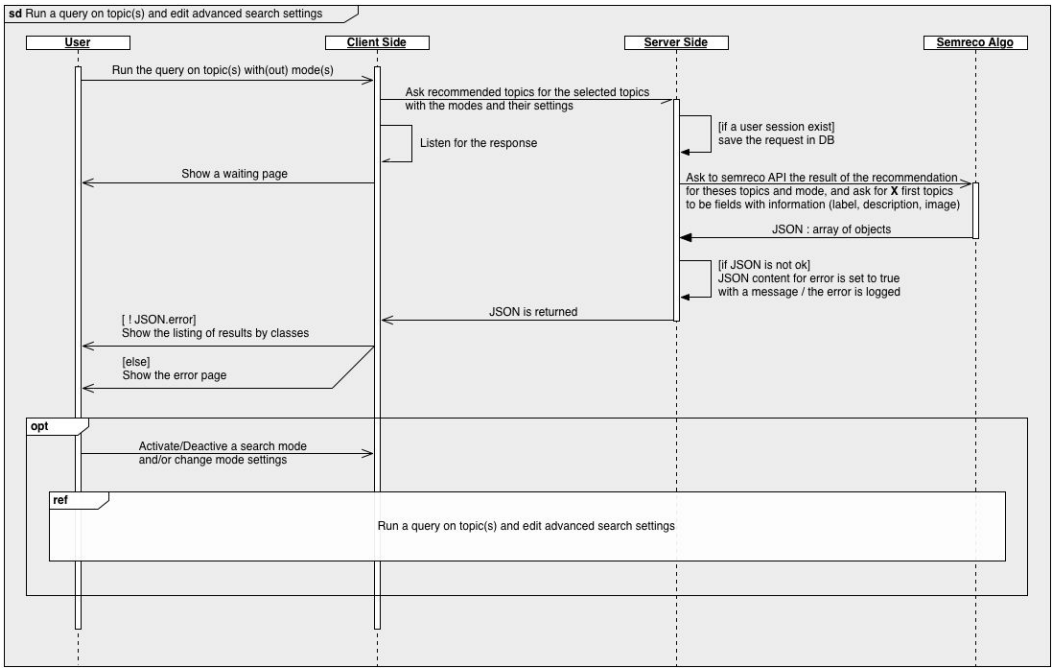

Figure E.1: Functional modeling of Discovery Hub V2: extract of an UML sequence diagram

APPENDIX F

Gantt chart extract

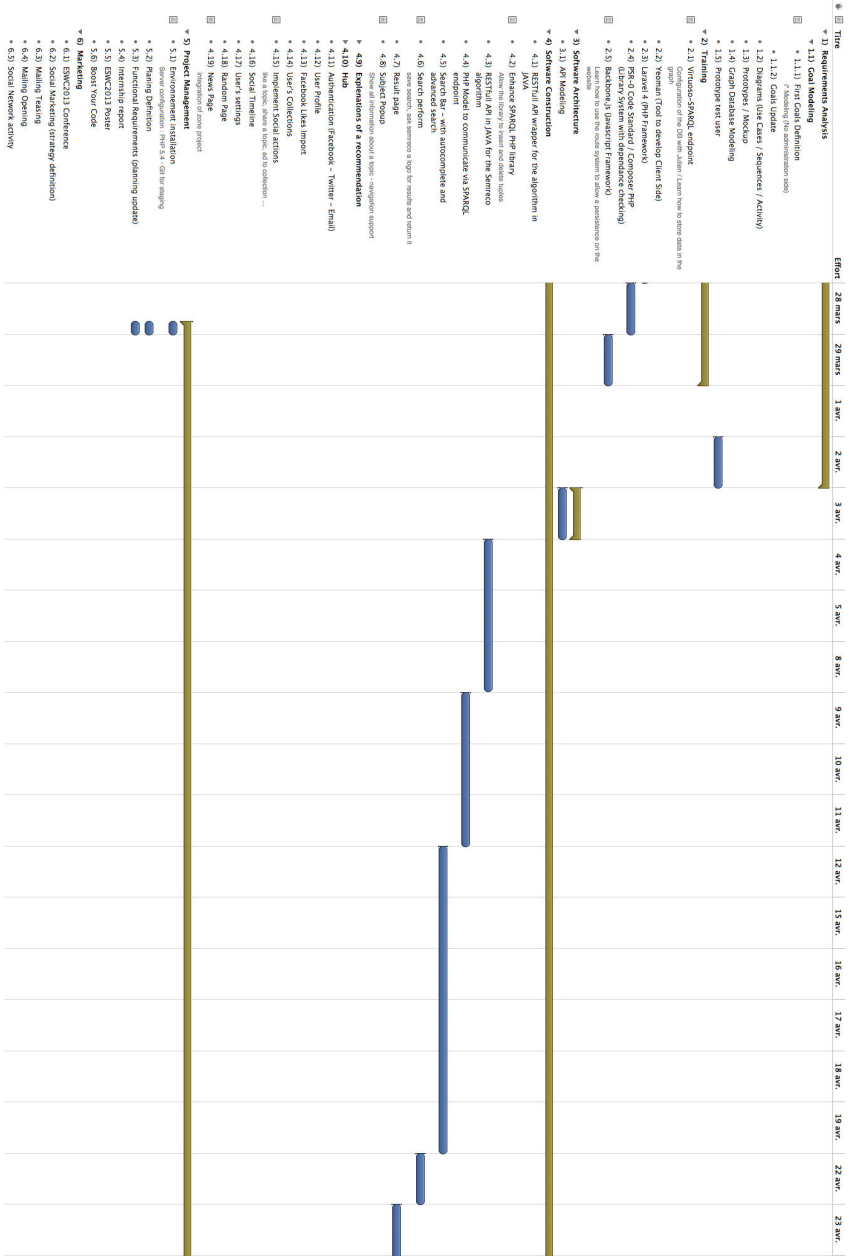

Figure F.1: Extract of a Gantt chart diagram used for the Discovery Hub V2 development

APPENDIX G

The Showcase Machine project, adapted from Discovery Hub

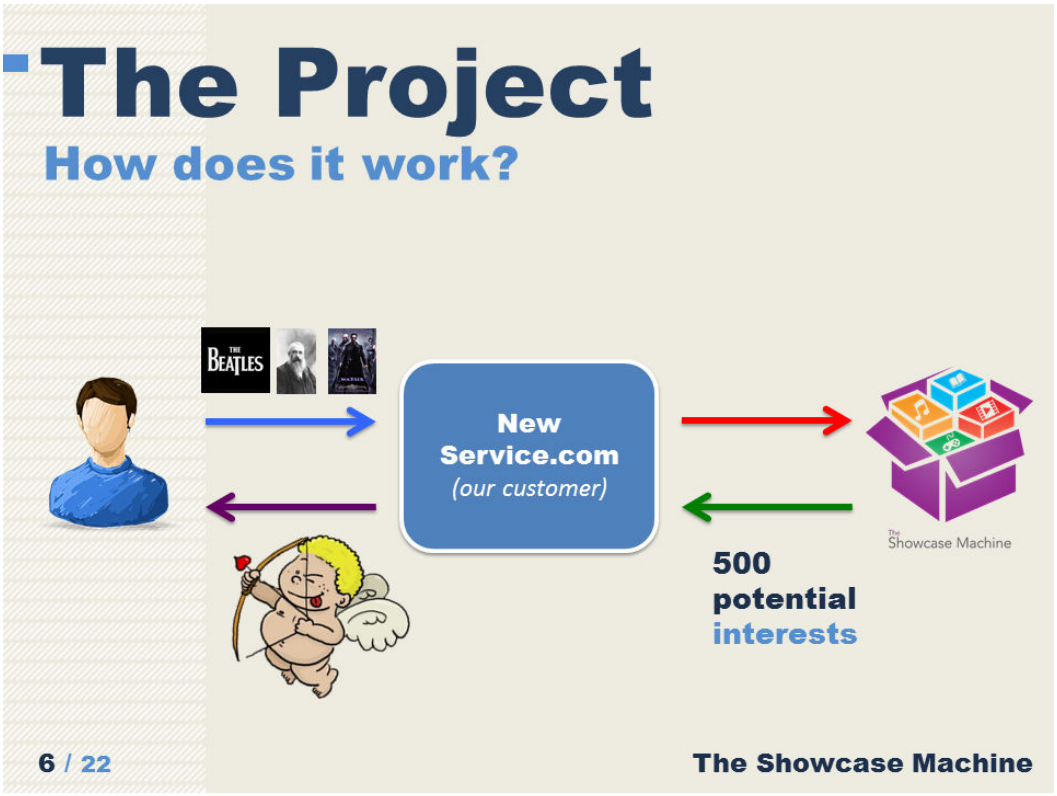

Figure G.1: A slide of the presentation for the finale of the Challenge Jeunes Pousses entrepreneurial challenge

APPENDIX H

ANOVA results for the monocentric queries evaluation

Factor	Measure	ANOVA Test
Algorithm	**Relevance**	F(1,14) = 113.85, p <.001
	Discovery	F(1,14) = 92.99, p <.001
Rank	**Relevance**	F(1,14) = 134.02, p <.001
	Discovery	F(1,14) = 64.30, p <.001
Algo * Rank	**Relevance**	F(1,14) = 20.23, p = .001
	Discovery	F(1,14) = 32.14, p <.001

Table H.1: Inferential statistics

Measure	Algo	Mean	Std. Error	95% Confidence Interval	
				Lower Bound	**Upper Bound**
Relevance	**1**	1,416	,070	1,266	1,566
	2	1,181	,063	1,046	1,317
Discovery	**1**	1,163	,059	1,035	1,290
	2	1,323	,055	1,205	1,442

Table H.2: Descriptive statistics - algorithm

Measure	Film	Mean	Std. Error	95% Confidence Interval	
				Lower Bound	**Upper Bound**
Relevance	**1**	1,228	,068	1,082	1,375
	2	1,018	,087	,831	1,205
	3	1,630	,065	1,491	1,769
	4	1,515	,099	1,302	1,728
	5	1,102	,086	,917	1,286
Discovery	**1**	1,087	,078	,919	1,254
	2	1,442	,073	1,286	1,598
	3	,882	,071	,729	1,034
	4	1,698	,071	1,547	1,850
	5	1,107	,079	,938	1,275

Table H.3: Descriptive statistics - film

Measure	Rang	Mean	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
Relevance	**1**	1,487	,076	1,323	1,651
	2	1,110	,058	,986	1,234
Discovery	**1**	1,125	,064	,987	1,262
	2	1,361	,053	1,248	1,474

Table H.4: Descriptive statistics - rank

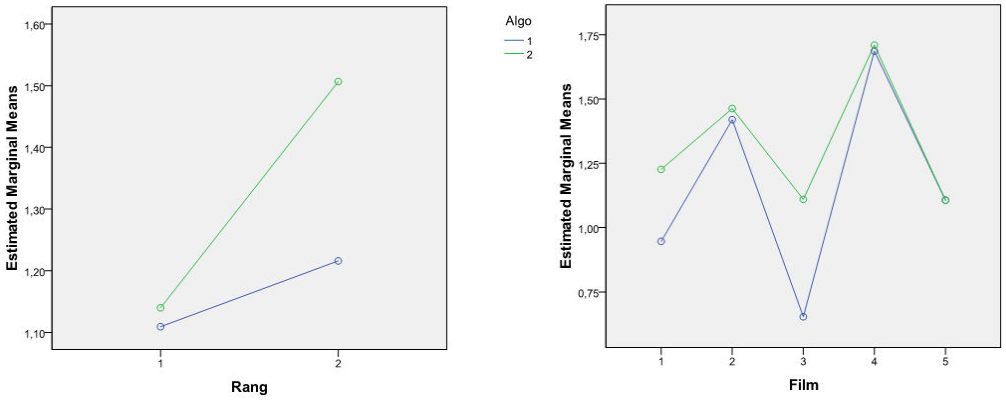

Figure H.1: Estimated marginal means of discovery by rank and by film

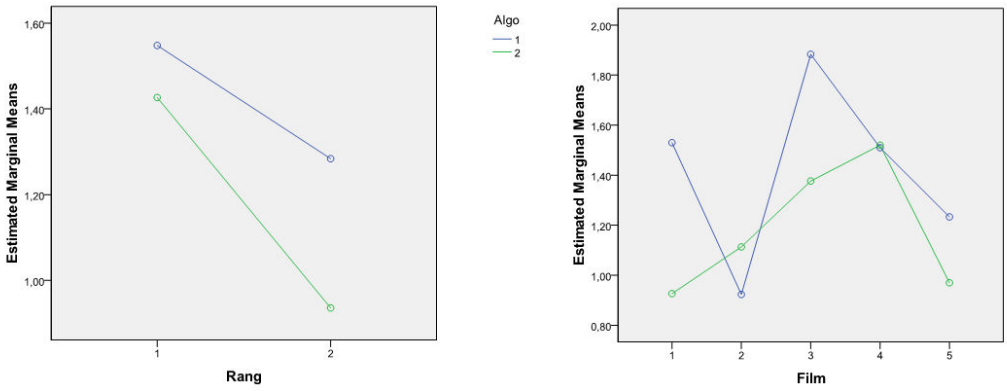

Figure H.2: Estimated marginal means of relevance by rank and by film

Measure	Algo	Film	Mean	Std. Error	95% Confidence Interval	
					Lower Bound	Upper Bound
Relevance	**1**	1	1,530	,072	1,376	1,684
		2	,923	,096	,717	1,129
		3	1,883	,078	1,717	2,050
		4	1,510	,112	1,270	1,750
		5	1,233	,092	1,035	1,432
	2	1	,927	,086	,743	1,111
		2	1,113	,084	,933	1,294
		3	1,377	,059	1,250	1,503
		4	1,520	,090	1,328	1,712
		5	,970	,083	,792	1,148
Discovery	**1**	1	,947	,078	,779	1,114
		2	1,420	,073	1,263	1,577
		3	,653	,086	,470	,837
		4	1,687	,077	1,522	1,851
		5	1,107	,089	,916	1,297
	2	1	1,227	,085	1,044	1,409
		2	1,463	,084	1,283	1,644
		3	1,110	,059	,983	1,237
		4	1,710	,067	1,566	1,854
		5	1,107	,073	,951	1,263

Table H.5: Descriptive statistics - algorithm * film

Measure	Algo	Rang	Mean	Std. Error	95% Confidence Interval	
					Lower Bound	Upper Bound
Relevance	**1**	**1**	1,548	,079	1,378	1,718
		2	1,284	,063	1,149	1,419
	2	**1**	1,427	,076	1,265	1,589
		2	,936	,059	,809	1,063
Discovery	**1**	**1**	1,109	,064	,971	1,248
		2	1,216	,059	1,090	1,342
	2	**1**	1,140	,065	1,002	1,278
		2	1,507	,053	1,393	1,620

Table H.6: Descriptive statistics - algorithm * rank

Measure	Film	Rang	Mean	Std. Error	95% Confidence Interval	
					Lower Bound	Upper Bound
Relevance	**1**	**1**	1,307	,086	1,122	1,492
		2	1,150	,060	1,022	1,278
	2	**1**	1,147	,090	,955	1,339
		2	,890	,090	,697	1,083
	3	**1**	2,043	,084	1,863	2,224
		2	1,217	,065	1,078	1,355
	4	**1**	1,750	,136	1,458	2,042
		2	1,280	,075	1,119	1,441
	5	**1**	1,190	,093	,991	1,389
		2	1,013	,081	,839	1,187
Discovery	**1**	**1**	,950	,085	,769	1,131
		2	1,223	,076	1,060	1,386
	2	**1**	1,353	,080	1,182	1,525
		2	1,530	,072	1,376	1,684
	3	**1**	,690	,087	,503	,877
		2	1,073	,062	,941	1,206
	4	**1**	1,623	,105	1,397	1,849
		2	1,773	,044	1,679	1,867
	5	**1**	1,007	,090	,813	1,201
		2	1,207	,083	1,028	1,385

Table H.7: Descriptive statistics - film * rank

APPENDIX I

Participants exploratory search-sessions modeling

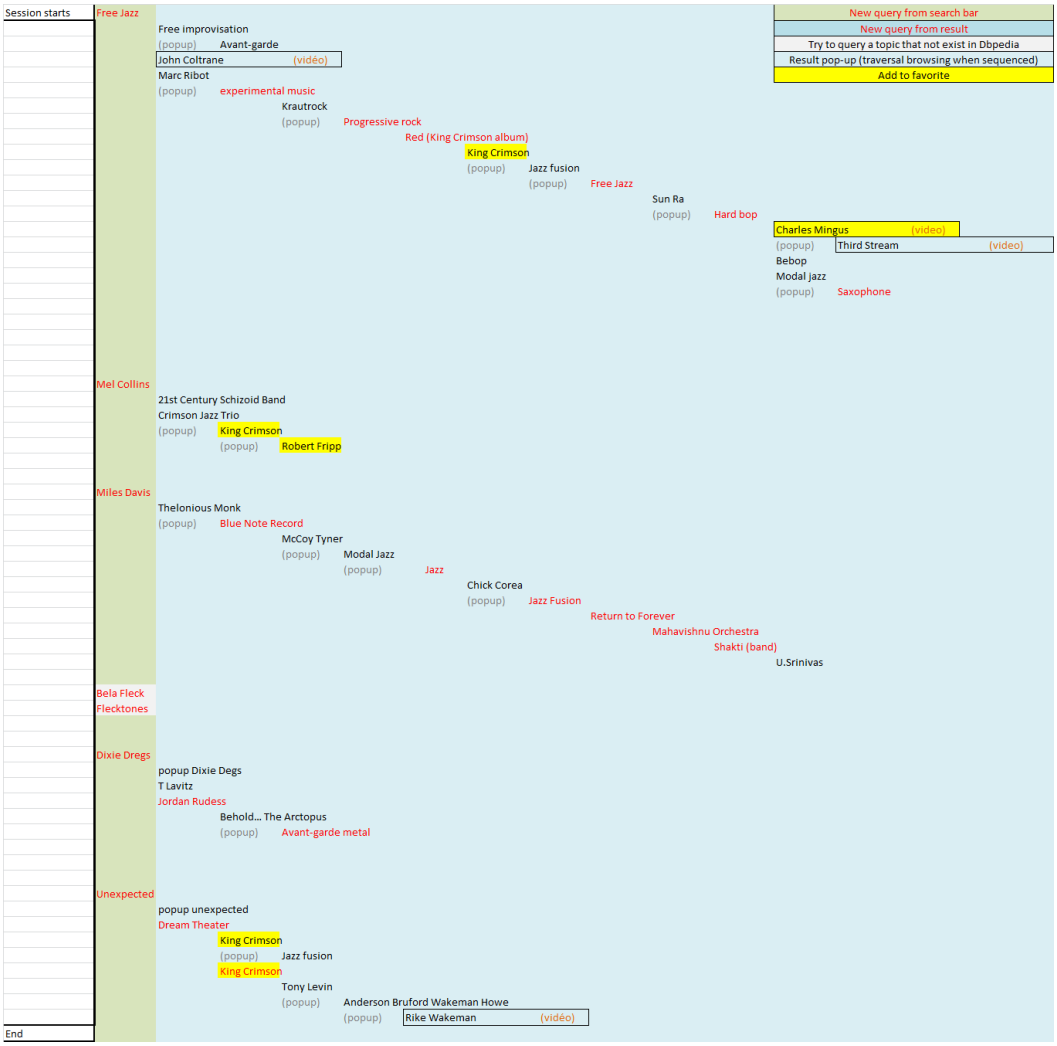

Figure I.1: Exploratory search-sessions modeling - participant 1

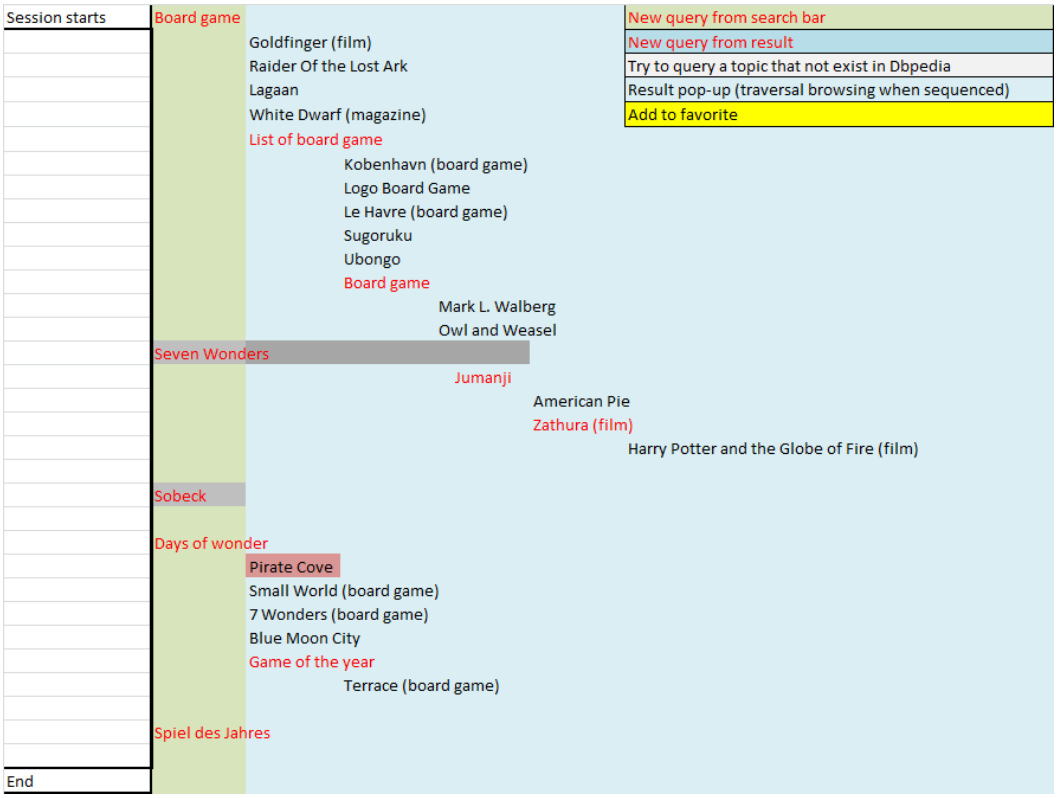

Figure I.2: Exploratory search-sessions modeling - participant 2

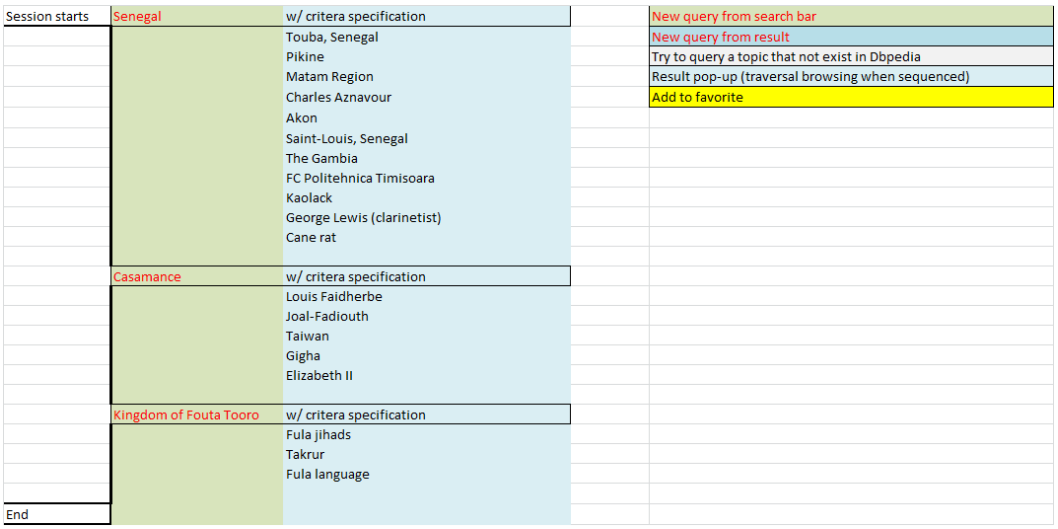

Figure I.3: Exploratory search-sessions modeling - participant 3

List of Figures

List of Tables

Bibliography

[1] N. M. Akim, A. Dix, A. Katifori, G. Lepouras, N. Shabir, and C. Vassilakis. Spreading activation for web scale reasoning: Promise and problems, 2011.

[2] W. Al Sarraj. A usability evaluation framework for web mashup makers for end-users. 2012.

[3] H. Alani, S. Dasmahapatra, K. O'Hara, and N. Shadbolt. Identifying communities of practice through ontology network analysis. *IEEE Intelligent Systems*, 18(2):18–25, 2003.

[4] J. R. Anderson. A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, 22:261–295, 1983.

[5] P. André, J. Teevan, S. T. Dumais, et al. Discovery is never by chance: designing for (un) serendipity. In *Proceedings of the seventh ACM conference on Creativity and cognition*, pages 305–314. ACM, 2009.

[6] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[7] S. Auer, R. Doehring, and S. Dietzold. Less-template-based syndication and presentation of linked data. In *The Semantic Web: Research and Applications*, pages 211–224. Springer, 2010.

[8] R. Baeza-Yates, M. Ciaramita, P. Mika, and H. Zaragoza. Towards semantic search. In *Natural Language and Information Systems*, pages 4–11. Springer, 2008.

[9] R. Baeza-Yates, A. Gionis, F. P. Junqueira, V. Murdock, V. Plachouras, and F. Silvestri. Design trade-offs for search engine caching. *ACM Transactions on the Web (TWEB)*, 2(4):20, 2008.

[10] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[11] M. Barbera. Linked (open) data at web scale: research, social and engineering challenges in the digital humanities. *JLIS. it*, 4(1):91, 2013.

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

[13] C. Becker and C. Bizer. Dbpedia mobile: A location-enabled linked data browser. *LDOW*, 369, 2008.

[14] T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop*, volume 2006, 2006.

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

[16] M. R. Berthold, U. Brandes, T. Kötter, M. Mader, U. Nagel, and K. Thiel. Pure spreading activation is pointless. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1915–1918. ACM, 2009.

[17] C. Bizer, K. Eckert, R. Meusel, H. Mühleisen, M. Schuhmacher, and J. Völker. Deployment of rdfa, microdata, and microformats on the web–a quantitative analysis. In *The Semantic Web–ISWC 2013*, pages 17–32. Springer, 2013.

[18] C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *International journal on semantic web and information systems*, 5(3):1–22, 2009.

[19] C. Bizer, P. N. Mendes, and A. Jentzsch. Topology of the web of data. In *Semantic Search over the Web*, pages 3–29. Springer, 2012.

[20] R. Blanco, B. B. Cambazoglu, P. Mika, and N. Torzec. Entity recommendations in web search. In *The Semantic Web–ISWC 2013*, pages 33–48. Springer, 2013.

[21] M. Brambilla and S. Ceri. Designing exploratory search applications upon web data sources. In *Semantic Search over the Web*, pages 61–77. Springer, 2012.

[22] R. L. Brennan. *Generalizability theory*. Springer, 2001.

[23] J. Breslin and S. Decker. The future of social networks on the internet: the need for semantics. *Internet Computing, IEEE*, 11(6):86–90, 2007.

[24] V. Bush. As we may think. 1945.

[25] R. Capra, G. Marchionini, J. S. Oh, F. Stutzman, and Y. Zhang. Effects of structure and interaction style on distinct search tasks. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 442–451. ACM, 2007.

[26] R. G. Capra and G. Marchionini. The relation browser tool for faceted exploratory search. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 420–420. ACM, 2008.

[27] S. Castano, A. Ferrara, and S. Montanelli. Thematic clustering and exploration of linked data. In *Search Computing*, pages 157–175. Springer, 2012.

[28] S. Castano, A. Ferrara, and S. Montanelli. inwalk: Interactive and thematic walks inside the web of data. In *EDBT*, pages 628–631, 2014.

[29] S. Ceri, A. Bozzon, M. Brambilla, E. Della Valle, P. Fraternali, and S. Quarteroni. Semantic search. In *Web Information Retrieval*, pages 181–206. Springer, 2013.

[30] P. R. Cohen and R. Kjeldsen. Information retrieval by constrained spreading activation in semantic networks. *Information processing & management*, 23(4):255–268, 1987.

[31] A. M. Collins and E. F. Loftus. A spreading activation theory of semantic processing. *Psychological Review*, 82:407–428, 1975.

[32] A. M. Collins and M. R. Quillian. Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior*, 8(2):240–247, 1969.

[33] O. Corby, R. Dieng-Kuntz, and C. Faron-Zucker. Querying the semantic web with corese search engine. In *ECAI*, volume 16, page 705, 2004.

[34] O. Corby, R. Dieng-Kuntz, C. Faron-Zucker, F. L. Gandon, et al. Ontology-based approximate query processing for searching the semantic web with corese. 2006.

[35] B. Craft and P. Cairns. Beyond guidelines: what can we learn from the visual information seeking mantra? In *Information Visualisation, 2005. Proceedings. Ninth International Conference on*, pages 110–118. IEEE, 2005.

[36] F. Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482, 1997.

[37] F. Crestani and P. L. Lee. Searching the web by constrained spreading activation. *Information Processing & Management*, 36(4):585–605, 2000.

[38] W. B. Croft. Approaches to intelligent information retrieval. *Information Processing & Management*, 23(4):249–254, 1987.

[39] M. Czarkowski and J. Kay. A scrutable adaptive hypertext. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 384–387. Springer, 2002.

[40] A.-S. Dadzie and M. Rowe. Approaches to visualising linked data: A survey. *Semantic Web*, 2(2):89–124, 2011.

[41] A.-S. Dadzie, M. Rowe, and D. Petrelli. Hide the stack: toward usable linked data. In *The Semantic Web: Research and Applications*, pages 93–107. Springer, 2011.

[42] N. Dalvi, R. Kumar, B. Pang, R. Ramakrishnan, A. Tomkins, P. Bohannon, S. Keerthi, and S. Merugu. A web of concepts. In *Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–12. ACM, 2009.

[43] T. Di Noia, R. Mirizzi, V. C. Ostuni, D. Romito, and M. Zanker. Linked open data to support content-based recommender systems. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 1–8. ACM, 2012.

[44] R. Dieng and S. Hug. Comparison of personal ontologies represented through conceptual graphs. In *ECAI*, volume 98, pages 341–345. Citeseer, 1998.

[45] A. Diriye, M. L. Wilson, A. Blandford, and A. Tombros. Revisiting exploratory search from the hci perspective. *HCIR 2010*, page 99, 2010.

[46] M. Duffy. Sensemaking in classroom conversations. *Openness in research: The tension between self and other*, pages 119–132, 1995.

[47] S. Dumais. Task-based search: A search engine perspective.

[48] J. Engestrom. Why some social network services work and others don't â or: the case for object-centered sociality, April 2005.

[49] J. Euzenat, P. Valtchev, et al. Similarity-based ontology alignment in owl-lite. In *ECAI*, volume 16, page 333, 2004.

[50] J. C. Fagan. Usability studies of faceted browsing: A literature review. *Information Technology and Libraries*, 29(2):58–66, 2013.

[51] I. Fernández-Tobías, I. Cantador, M. Kaminskas, and F. Ricci. A generic semantic-based framework for cross-domain recommendation. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pages 25–32. ACM, 2011.

[52] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010.

[53] A. Freitas, J. G. Oliveira, E. Curry, S. OâRiain, and J. C. P. da Silva. Treo: combining entity-search, spreading activation and semantic relatedness for querying linked data. In *Proc. of 1st Workshop on Question Answering over Linked Data (QALD-1) at the 8th Extended Semantic Web Conference (ESWC 2011)*, 2011.

[54] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[55] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

[56] A. Gangemi, A. G. Nuzzolese, V. Presutti, F. Draicchio, A. Musetti, and P. Ciancarini. Automatic typing of dbpedia entities. In *The Semantic Web–ISWC 2012*, pages 65–81. Springer, 2012.

[57] M. S. Gazzaniga, R. Ivry, and G. Mangun. Fundamentals of cognitive neuroscience, 1998.

[58] G. Golovchinsky, J. Adcock, J. Pickens, P. Qvarfordt, and M. Back. Cerchiamo: a collaborative exploratory search tool. *Proceedings of Computer Supported Cooperative Work (CSCW)*, 2008.

[59] S. Gouws, G. Van Rooyen, and H. A. Engelbrecht. Measuring conceptual similarity by spreading activation over wikipediaâs hyperlink structure. In *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, page 46, 2010.

[60] S. Goyal and R. Westenthaler. Rdf gravity (rdf graph visualization tool). *Salzburg Research, Austria*, 2004.

[61] S. Greene, G. Marchionini, C. Plaisant, and B. Shneiderman. Previews and overviews in digital libraries: Designing surrogates to support visual information seeking. *Journal of the American Society for Information Science*, 51(4):380–393, 2000.

[62] M. Grinberg, V. Haltakov, and H. Stefanov. Approximate spreading activation for efficient knowledge retrieval from large datasets. In *Proceeding of the 2011 conference on Neural Nets WIRN10: Proceedings of the 20th Italian Workshop on Neural Nets*, page 326, 2011.

[63] V. Groues, Y. Naudet, and O. Kao. Adaptation and evaluation of a semantic similarity measure for dbpedia: A first experiment. In *Semantic and Social Media Adaptation and Personalization (SMAP), 2012 Seventh International Workshop on*, pages 87–91. IEEE, 2012.

[64] N. Guarino and P. Giaretta. *Ontologies and Knowledge Bases - Towards a Terminological Clarification*, pages 25–32. IOS Press, Amsterdam, The Netherlands, 1995.

[65] R. Guha, R. McCool, and E. Miller. Semantic search. In *Proceedings of the 12th international conference on World Wide Web*, pages 700–709. ACM, 2003.

[66] K. Haas, P. Mika, P. Tarjan, and R. Blanco. Enhanced results for web search. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 725–734. ACM, 2011.

[67] S. W. Haas. Text-based intelligent systems: Current research and practice in information extraction and retrieval. *Associates*, 8:281, 1992.

[68] R. Hahn, C. Bizer, C. Sahnwaldt, C. Herta, S. Robinson, M. Bürgle, H. Düwiger, and U. Scheel. Faceted wikipedia search. In *Business Information Systems*, pages 1–11. Springer, 2010.

[69] D. Hardman and L. Edwards. âlost in hyperspaceâ: Cognitive mapping and navigation in a hypertext environment. *Hypertext: Theory into practice*, pages 105–145, 1989.

[70] A. Harth. Visinav: A system for visual search and navigation on web data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4):348–354, 2010.

[71] T. Hastrup, R. Cyganiak, and U. Bojars. Browsing linked data with fenfire. 2008.

[72] M. A. Hearst. Next generation web search: Setting our sites. *IEEE Data Engineering Bulletin*, 23(3):38–48, 2000.

[73] P. Heim, T. Ertl, and J. Ziegler. Facet graphs: Complex semantic querying made easy. In *The Semantic Web: Research and Applications*, pages 288–302. Springer, 2010.

[74] P. Heim, S. Hellmann, J. Lehmann, S. Lohmann, and T. Stegemann. Relfinder: Revealing relationships in rdf knowledge bases. In *Semantic Multimedia*, pages 182–187. Springer, 2009.

[75] P. Heim, J. Ziegler, and S. Lohmann. gfacet: A browser for the web of data. In *Proceedings of the International Workshop on Interacting with Multimedia Content in the Social Semantic Web (IMC-SSWâ08)*, volume 417, pages 49–58. Citeseer, 2008.

[76] B. Heitmann. An open framework for multi-source, cross-domain personalisation with semantic interest graphs. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 313–316. ACM, 2012.

[77] B. Heitmann and C. Hayes. Report on evaluation of recommendations generated by spreading activation. 2013.

[78] M. Hildebrand, J. van Ossenbruggen, and L. Hardman. /facet: A browser for heterogeneous semantic web repositories. In *The Semantic Web-ISWC 2006*, pages 272–285. Springer, 2006.

[79] T. Hussein and J. Ziegler. Adapting web sites by spreading activation in ontologies. In *Proceedings of International Workshop on Recommendation and Collaboration, New York, USA*, 2008.

[80] D. Huynh, S. Mazzocchi, and D. Karger. Piggy bank: Experience the semantic web inside your web browser. In *The Semantic Web–ISWC 2005*, pages 413–430. Springer, 2005.

[81] D. F. Huynh and D. Karger. Parallax and companion: Set-based browsing for the data web. In *WWW Conference. ACM*. Citeseer, 2009.

[82] D. F. Huynh, D. R. Karger, and R. C. Miller. Exhibit: lightweight structured data publishing. In *Proceedings of the 16th international conference on World Wide Web*, pages 737–746. ACM, 2007.

[83] E. Hyvönen, E. Mäkelä, M. Salminen, A. Valo, K. Viljanen, S. Saarela, M. Junnila, and S. Kettula. Museumfinlandâfinnish museums on the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2):224–241, 2005.

[84] M. Käki. Findex: search result categories help users when document ranking fails. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 131–140. ACM, 2005.

[85] M. Kaminskas, I. Fernández-Tobías, I. Cantador, and F. Ricci. *Ontology-based identification of music for places*. Springer, 2013.

[86] M. Kaminskas, I. Fernández-Tobías, F. Ricci, and I. Cantador. Knowledge-based music retrieval for places of interest. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, pages 19–24. ACM, 2012.

[87] M. Kaminskas and F. Ricci. Location-adapted music recommendation using tags. In *User Modeling, Adaption and Personalization*, pages 183–194. Springer, 2011.

[88] Y. Kammerer, R. Nairn, P. Pirolli, and E. H. Chi. Signpost from the masses: learning effects in an exploratory social tag search browser. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 625–634. ACM, 2009.

[89] D. Karger et al. The pathetic fallacy of rdf. 2006.

[90] A. Katifori, C. Vassilakis, and A. Dix. Ontologies and the brain: Using spreading activation through ontologies to support personal interaction. *Cognitive Systems Research*, 11(1):25–41, 2010.

[91] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. *Visual analytics: Definition, process, and challenges*. Springer, 2008.

[92] M. G. Kendall. Rank correlation methods (london: Charles griffin, 1948). *KendallRank Correlation Methods1948*, 1970.

[93] S. Kinsella, A. Harth, A. Troussov, M. Sogrin, J. Judge, C. Hayes, and J. G. Breslin. Navigating and annotating semantically-enabled networks of people and associated objects. In *Why Context Matters*, pages 79–96. Springer, 2008.

[94] G. Klein, B. M. Moon, and R. R. Hoffman. Making sense of sensemaking 1: Alternative perspectives. *IEEE intelligent systems*, 21(4):70–73, 2006.

[95] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[96] G. Kobilarov and I. Dickinson. Humboldt: Exploring linked data. *context*, 6:7, 2008.

[97] J. Koch, T. Franz, and S. Staab. Lena-browsing rdf data more complex than foaf. In *International Semantic Web Conference (Posters & Demos)*, 2008.

[98] J. Koren, Y. Zhang, and X. Liu. Personalized interactive faceted search. In *Proceedings of the 17th international conference on World Wide Web*, pages 477–486. ACM, 2008.

[99] C. Koumenides. *A Bayesian network model for entity-oriented semantic web search*. PhD thesis, University of Southampton, 2013.

[100] C. L. Koumenides and N. R. Shadbolt. Combining link and content-based information in a bayesian inference model for entity search. In *Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search*, page 3. ACM, 2012.

[101] C. L. Koumenides and N. R. Shadbolt. Ranking methods for entity-oriented semantic web search. *Journal of the Association for Information Science and Technology*, 65(6):1091–1106, 2014.

[102] B. Kules, R. Capra, M. Banta, and T. Sierra. What do exploratory searchers look at in a faceted search interface? In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 313–322. ACM, 2009.

[103] B. Kules and B. Shneiderman. Users can change their web search tactics: Design guidelines for categorized overviews. *Information Processing & Management*, 44(2):463–484, 2008.

[104] W. M. Kules III and B. Adviser-Shneiderman. *Supporting exploratory web search with meaningful and stable categorized overviews*. University of Maryland at College Park, 2006.

[105] J. Kunegis. Konect: the koblenz network collection. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 1343–1350. International World Wide Web Conferences Steering Committee, 2013.

[106] B. H. Kwasnik. A descriptive study of the functional components of browsing. In *Proceedings of the IFIP TC2/WG2. 7 Working conference on Engineering for Human Computer Interaction*, volume 18, pages 191–203, 1992.

[107] K. La Barre. Facet analysis. *Annual Review of Information Science and Technology*, 44(1):243–284, 2010.

[108] S. Lam, C. Hayes, N. DERI, and I. B. Park. Using the structure of dbpedia for exploratory search.

[109] M. Lee, W. Kim, and T. Wang. An explorative association-based search for the semantic web. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pages 206–211. IEEE, 2010.

[110] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2006.

[111] T.-P. Liang, Y.-F. Yang, D.-N. Chen, and Y.-C. Ku. A semantic-expansion approach to personalized knowledge recommendation. *Decision Support Systems*, 45(3):401–412, 2008.

[112] R. Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[113] T. Lin, P. Pantel, M. Gamon, A. Kannan, and A. Fuxman. Active objects: Actions for entity-centric search. In *Proceedings of the 21st international conference on World Wide Web*, pages 589–598. ACM, 2012.

[114] W. Liu, A. Weichselbraun, A. Scharl, and E. Chang. Semi-automatic ontology extension using spreading activation. *Journal of Universal Knowledge Management*, 1:50–58, 2005.

[115] A. Loizou. *How to recommend music to film buffs: enabling the provision of recommendations from multiple domains*. PhD thesis, University of Southampton, 2009.

[116] J. G. Lorenzo, J. E. L. Gayo, and J. M. Á. Rodríguez. Applying mapreduce to spreading activation algorithm on large rdf graphs. In *Information Systems, E-learning, and Knowledge Management Research*, pages 601–611. Springer, 2013.

[117] A. Maedche and V. Zacharias. Clustering ontology-based metadata in the semantic web. In *Principles of Data Mining and Knowledge Discovery*, pages 348–360. Springer, 2002.

[118] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 135–146. ACM, 2010.

[119] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[120] G. Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[121] G. Marchionini. Toward human-computer information retrieval. *BULLETIN-AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY*, 32(5):20, 2006.

[122] G. Marchionini and R. W. White. Information-seeking support systems. *IEEE Computer*, 42(3):30–32, 2009.

[123] N. Marie, O. Corby, F. Gandon, and M. Ribière. Composite interests' exploration thanks to on-the-fly linked data spreading activation. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pages 31–40. ACM, 2013.

[124] N. Marie and F. Gandon. Demonstration of multi-perspectives exploratory search with the discovery hub web application. In *ISWC*, 2014.

[125] N. Marie, F. Gandon, et al. Survey of linked data based exploration systems. In *IESD 2014-Intelligent Exploitation of Semantic Data*, 2015.

[126] N. Marie, F. Gandon, A. Giboin, É. Palagi, et al. Exploratory search on topics through different perspectives with dbpedia. In *SEMANTICS*, 2014.

[127] N. Marie, F. Gandon, D. Legrand, and M. Ribière. Exploratory search on the top of dbpedia chapters with the discovery hub application. In *The Semantic Web: ESWC 2013 Satellite Events*, pages 184–188. Springer, 2013.

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

[129] D. L. McCuinness. Ontologies come of age. *Spinning the semantic web: bringing the World Wide Web to its full potential*, page 171, 2005.

[130] R. Meymandpoura and J. G. Davisa. Similarity metrics for linked data.

[131] D. S. E. Mirizzi, Roberto and T. Di Noia. Exploratory search and recommender systems in the semantic web.

[132] R. Mirizzi and T. Di Noia. From exploratory search to web search and back. In *Proceedings of the 3rd workshop on Ph. D. students in information and knowledge management*, pages 39–46. ACM, 2010.

[133] R. Mirizzi, T. Di Noia, A. Ragone, V. C. Ostuni, and E. Di Sciascio. Movie recommendation with dbpedia. In *IIR*, pages 101–112. Citeseer, 2012.

[134] R. Mirizzi, A. Ragone, T. Di Noia, and E. Di Sciascio. *Ranking the linked data: the case of dbpedia*. Springer, 2010.

[135] R. Mirizzi, A. Ragone, T. Di Noia, and E. Di Sciascio. *Semantic wonder cloud: exploratory search in DBpedia*. Springer, 2010.

[136] M. R. Morris. Interfaces for collaborative exploratory web search: Motivations and directions for multi-user design. In *Proceedings of ACM SIGCHI 2007 Workshop on Exploratory Search and HCI: Designing and Evaluating Interfaces to Support Exploratory Search Interaction*, pages 9–12, 2007.

[137] A. Musetti, A. G. Nuzzolese, F. Draicchio, V. Presutti, E. Blomqvist, A. Gangemi, and P. Ciancarini. Aemoo: Exploratory search based on knowledge patterns over the semantic web. *Semantic Web Challenge*, 2012.

[138] G. Neumann and S. Schmeier. Guided exploratory search on the mobile web.

[139] A. Newell, H. A. Simon, et al. *Human problem solving*, volume 104. Prentice-Hall Englewood Cliffs, NJ, 1972.

[140] A. G. Nuzzolese, A. Gangemi, V. Presutti, and P. Ciancarini. Encyclopedic knowledge patterns from wikipedia links. In *The Semantic Web–ISWC 2011*, pages 520–536. Springer, 2011.

[141] V. L. O'Day and R. Jeffries. Orienteering in an information landscape: how information seekers get from here to there. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 438–445. ACM, 1993.

[142] E. Oren, R. Delbru, and S. Decker. Extending faceted navigation for rdf data. In *The Semantic Web-ISWC 2006*, pages 559–572. Springer, 2006.

[143] V. C. Ostuni, G. Gentile, T. Di Noia, R. Mirizzi, D. Romito, and E. Di Sciascio. Mobile movie recommendations with linked data. In *Availability, Reliability, and Security in Information Systems and HCI*, pages 400–415. Springer, 2013.

[144] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[145] C. R. Palmer, G. Siganos, M. Faloutsos, C. Faloutsos, and P. B. Gibbons. The connectivity and fault-tolerance of the internet topology. 2001.

[146] E. Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011.

[147] A. Passant. Dbrecâmusic recommendations using dbpedia. In *The Semantic Web–ISWC 2010*, pages 209–224. Springer, 2010.

[148] A. Passant. Measuring semantic distance on linking data and using it for resources recommendations. In *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*, 2010.

[149] M. C. Pattuelli, M. Miller, L. Lange, S. Fitzell, and C. Li-Madeo. Crafting linked open data for cultural heritage: Mapping and curation tools for the linked jazz project. *Code4Lib Journal*, (21), 2013.

[150] J. Pickens, G. Golovchinsky, C. Shah, P. Qvarfordt, and M. Back. Algorithmic mediation for collaborative exploratory search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 315–322. ACM, 2008.

[151] P. L. Pirolli and J. R. Anderson. The role of practice in fact retrieval. *Journal of experimental psychology: Learning, memory, and cognition*, 11(1):136, 1985.

[152] I. O. Popov, M. Schraefel, W. Hall, and N. Shadbolt. Connecting the dots: a multi-pivot approach to data exploration. In *The Semantic Web–ISWC 2011*, pages 553–568. Springer, 2011.

[153] W. Pratt, M. A. Hearst, and L. M. Fagan. A knowledge-based approach to organizing retrieved documents. In *AAAI/IAAI*, pages 80–85, 1999.

[154] S. E. Preece. A spreading activation network model for information retrieval. 1981.

[155] E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF, 2005.

[156] Y. Qu and G. W. Furnas. Model-driven formative evaluation of exploratory search: A study under a sensemaking framework. *Information Processing & Management*, 44(2):534–555, 2008.

[157] M. R. Quillan. Semantic memory. Technical report, DTIC Document, 1966.

[158] S. R. Ranganathan and B. Palmer. *Elements of library classification*. Association of Assistant librarians London, 1959.

[159] C. Rocha, D. Schwabe, and M. P. de Aragão. A hybrid approach for searching in the semantic web. In S. I. Feldman, M. Uretsky, M. Najork, and C. E. Wills, editors, *WWW*, pages 374–383. ACM, 2004.

[160] M. A. Rodríguez and M. J. Egenhofer. Determining semantic similarity among entity classes from different ontologies. *Knowledge and Data Engineering, IEEE Transactions on*, 15(2):442–456, 2003.

[161] T. Ruotsalo, K. Athukorala, D. Głowacka, K. Konyushkova, A. Oulasvirta, S. Kaipiainen, S. Kaski, and G. Jacucci. Supporting exploratory search tasks with interactive user modeling. *Proc. ASIS&Tâ13*, 2013.

[162] L. Rutledge, J. Van Ossenbruggen, and L. Hardman. Making rdf presentable: integrated global and local semantic web browsing. In *Proceedings of the 14th international conference on World Wide Web*, pages 199–206. ACM, 2005.

[163] G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[164] J. Savoy. Bayesian inference networks and spreading activation in hypertext systems. *Information processing & management*, 28(3):389–406, 1992.

[165] J. B. Schafer, J. A. Konstan, and J. Riedl. E-commerce recommendation applications. In *Applications of Data Mining to Electronic Commerce*, pages 115–153. Springer, 2001.

[166] B. Shao, H. Wang, and Y. Li. The trinity graph engine. *Microsoft Research*, page 54, 2012.

[167] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[168] P. Shoval. Expert/consultation system for a retrieval data-base with semantic network of concepts. In *ACM SIGIR Forum*, volume 16, pages 145–149. ACM, 1981.

[169] J. Smith and B. Kules. Toward a design space for categorized overviews of search results. *Retrieved September*, 27:2007, 2006.

[170] D. Soergel. The rise of ontologies or the reinvention of classification. *JASIS*, 50(12):1119–1120, 1999.

[171] R. C. Sprinthall and S. T. Fisk. *Basic statistical analysis*. Prentice Hall Englewood Cliffs, NJ, 1990.

[172] J. Stan. *A semantic framework for social search*. PhD thesis, Saint Etienne, 2011.

[173] M. Stankovic. *Convergence entre Web Social et Web Sémantique. Application à l'innovation à l'aide du Web*. PhD thesis, Paris 4, 2012.

[174] M. Stankovic, W. Breitfuss, and P. Laublet. Linked-data based suggestion of relevant topics. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 49–55. ACM, 2011.

[175] T. Steiner and S. Mirea. Seki@ home, or crowdsourcing an open knowledge graph. In *Proceedings of the First International Workshop on Knowledge Extraction and Consolidation from Social Media (KECSM2012), Boston, USA*, 2012.

[176] E. Stoica and M. A. Hearst. Nearly-automated metadata hierarchy creation. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 117–120. Association for Computational Linguistics, 2004.

[177] O. Suominen and E. Hyvönen. Improving the quality of skos vocabularies with skosify. In *Knowledge Engineering and Knowledge Management*, pages 383–397. Springer, 2012.

[178] J. Teevan, K. Collins-Thompson, R. W. White, S. T. Dumais, and Y. Kim. Slow search: Information retrieval without time constraints. In *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval*, page 1. ACM, 2013.

[179] D. Torres, P. Molli, H. Skaf-Molli, and A. Diaz. Improving wikipedia with dbpedia. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 1107–1112. ACM, 2012.

[180] T. Tran and P. Mika. A survey of semantic search approaches.

[181] G. Tummarello, R. Cyganiak, M. Catasta, S. Danielczyk, R. Delbru, and S. Decker. Sig. ma: Live views on the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4):355–364, 2010.

[182] W. Tunstall-Pedoe. True knowledge: Open-domain question answering using structured knowledge and inference. *AI Magazine*, 31(3):80–92, 2010.

[183] A. Tversky. Features of similarity. *Psychological review*, 84(4):327, 1977.

[184] M. N. Uddin and P. Janecek. Performance and usability testing of multidimensional taxonomy in web site search and navigation. *Performance measurement and metrics*, 8(1):18–33, 2007.

[185] M. Uschold and M. Gruninger. Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11(02):93–136, 1996.

[186] M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer. Semantic wikipedia. In *Proceedings of the 15th international conference on World Wide Web*, pages 585–594. ACM, 2006.

[187] D. Vrandecic and M. Krötzsch. Wikidata: a free collaborative knowledge base. *Communications of the ACM*, 2014.

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

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

[190] J. Waitelonis, N. Ludwig, M. Knuth, and H. Sack. Whoknows? evaluating linked data heuristics with a quiz that cleans up dbpedia. *Interactive Technology and Smart Education*, 8(4):236–248, 2011.

[191] J. Waitelonis and H. Sack. Towards exploratory video search using linked data. *Multimedia Tools and Applications*, 59(2):645–672, 2012.

[192] R. W. White, B. Kules, S. M. Drucker, et al. Supporting exploratory search, introduction, special issue, communications of the acm. *Communications of the ACM*, 49(4):36–39, 2006.

[193] R. W. White, G. Marchionini, and G. Muresan. Evaluating exploratory search systems: Introduction to special topic issue of information processing and management. *Information Processing & Management*, 44(2):433–436, 2008.

[194] 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.

[195] B. M. Wildemuth and L. Freund. Assigning search tasks designed to elicit exploratory search behaviors. In *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval*, page 4. ACM, 2012.

[196] M. Wilson, A. Russell, D. A. Smith, et al. Mspace mobile: A ui gestalt to support on-the-go info-interaction. In *CHI'06 extended abstracts on Human factors in computing systems*, pages 247–250. ACM, 2006.

[197] M. L. Wilson and D. Elsweiler. Casual-leisure searching: the exploratory search scenarios that break our current models. 2010.

[198] M. L. Wilson et al. Improving exploratory search interfaces: Adding value or information overload? 2008.

[199] M. L. Wilson et al. A longitudinal study of exploratory and keyword search. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 52–56. ACM, 2008.

[200] M. L. Wilson, A. Russell, D. A. Smith, A. Owens, et al. mspace mobile: A mobile application for the semantic web. 2005.

[201] M. L. Wilson, R. W. White, et al. Evaluating advanced search interfaces using established information-seeking models. *Journal of the American Society for Information Science and Technology*, 60(7):1407–1422, 2009.

[202] P. Winoto and T. Tang. If you like the devil wears prada the book, will you also enjoy the devil wears prada the movie? a study of cross-domain recommendations. *New Generation Computing*, 26(3):209–225, 2008.

[203] I. Witten and D. Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, pages 25–30, 2008.

[204] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408. ACM, 2003.

[205] E. Yeh, D. Ramage, C. D. Manning, E. Agirre, and A. Soroa. Wikiwalk: random walks on wikipedia for semantic relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 41–49. Association for Computational Linguistics, 2009.

[206] J. Zhang and G. Marchionini. Evaluation and evolution of a browse and search interface: Relation browser++. In *Proceedings of the 2005 national conference on Digital government research*, pages 179–188. Digital Government Society of North America, 2005.

[207] M. Zollinger, C. Basca, and A. Bernstein. Market-based sparql brokerage: Towards economic incentives for linked data growth. In *The Semantic Web: ESWC 2013 Satellite Events*, pages 282–284. Springer, 2013.