Frameworks for Information Exploration – A Case Study

Thiago Nunes, Daniel Schwabe

Department of Informatics
Pontifical Catholic University of Rio de Janeiro
R.M.S. Vicente 225
Gávea Rio de Janeiro, RJ, Brazil
+55 21 3527-1500

{tnunes, dschwabe}@inf.puc-rio.br

Abstract. The exploration of information has become a common task among data consumers, leveraged mostly by the increasing availability of (semi) structured data on the Web. Even though there has been much research on data visualization techniques to support sense making over large datasets, the design and development of tools to support exploration actions with sufficient expressivity still lacks a systematic approach. In this paper we present a case study as a means to argue the relevance of having a model of operations as approach to address expressivity issues in exploration tools. The case study demonstrates the value of a framework of operations not only for tool comparisons, but also for representing, describing, and leveraging sharing of exploration patterns among communities of users.

1 Introduction

The availability of Structured Data on the Web has become both an opportunity and a challenge [9]. Parallel to the emergence of data stored in traditional relational databases, the availability of Linked Open Data in the WWW has also increased tremendously [11]. Once relevant data has been found, a number of environments can help the designer to clean up, aggregate, and present the data to users through rich interfaces, such as maps and infographics.

Exploratory Search is usually considered as an information problem that pushes the range of tasks from isolated sequences of query-response interactions to integrated searching and browsing activities aiming at learning [21, 31]. The outcome of exploratory searches is usually the knowledge achieved from the analysis of a set of items and its structure. Nevertheless, the sequence of actions that led to the desired knowledge is not considered as a relevant outcome, worth representing in a formal way that can leverage discovery and sharing of exploration patterns. We adopt here the term *Information Exploration* to refer to a range of tasks and actions that goes beyond searching and browsing a collection of items and also involves management of the knowledge regarding the process itself, including the reuse and sharing of exploration steps and patterns, preferably leveraged by a formal exploration model.

Information Exploration have been previously employed to designate the process of refinement of a vague information need by interacting with information objects

[30], aiming at knowledge acquisition within a defined conceptual area [5]. Furthermore, Information Exploration is also considered as a broad class of activities having Exploratory Search as a specialization [31], which, motivated the adoption by this work.

Since *Information Exploration* tasks are an extension of the usual Exploratory Search concept, they share many characteristics, such as, having a inherent considerable degree of complexity, being composed by a sequence of actions carried out on multiple information items [32], and being motivated mostly by lack of a-priori knowledge [21] and often plain curiosity [31].

Even though there has been much research on exploration tools and visualization techniques [9, 19, 20, 26], these works do not present effective approaches allowing to leverage the separation of concerns – interface/interaction and functionality in data manipulation - in Information Exploration. As a consequence, it is hard to assess both how adequate a tool is for a given exploration task and for which kinds of tasks the tool provides sufficient support. In a previous work [23], we argued the benefits of applying the separation of concerns principle to separate interface and interaction issues from the underlying conceptual model of exploration actions and strategies. We also argued for the construction of a framework of exploration operations as a unified way for discussing exploration primitives and comparing tools, leveraging formal descriptions of exploration solutions, and sharing of exploration patterns. In this work, we extend the previous work demonstrating through a case study on patent analysis how the framework allows the description of exploration tasks, tool comparisons, and generalization and reuse of solutions.

The organization of this work is as follows: section 2 introduces the main open research questions on the Information Exploration field and how we approach each question. Section 3 presents the framework we devised for addressing exploration concerns. Section 4 describes the case study and the findings. Section 5 presents the conclusions and future works.

2 Research Questions

Although the exploration phenomenon has been studied along the last decade under the concept of "Exploratory Search", no conclusions have been drawn with regards to a sufficient set of actions involved in the process. Therefore, the central question addressed by this research is: what can be considered a sufficient enough set of primitive exploration actions that can support most exploration tasks?

Considering that exploration actions can be modeled as data manipulation operations, more specific questions arise. First, which operations are involved in an exploration process? What are their parameters? What are the results of their application to a set of items? Second, an exploration is accomplished by a sequence of actions where the results of previous actions can be used as input to subsequent actions, hence forming functional compositions of operations. Thus, which compositions can be formed? For example, is it possible to issue a query over the results of a refinement operation?

By providing answers to these questions we expect to build a unified framework of exploration operations, which we believe can benefit the whole area of information exploration in the following ways:

- It can aggregate the knowledge and findings concerning data manipulation operations in exploration tasks in a common framework of operations;
- It can be used as the building blocks to compose complex task solutions. These compositions can serve as an analytical tool for analyzing the degree of expressivity of exploration tools, thereby separating interface design issues from exploration actions;
- The compositions can be used to discover and leverage exploration patterns among communities of users. As an example, some exploration tasks are recurrent among the community of patent analysts, such as, tracing changes in patent trends over two time periods [28], or discovering relationships between competitors based on their patenting behavior [22]. Furthermore, such patterns can, at a second stage, be shared among users, as well as generalized.

A precise and definitive answer to the central question is still an ongoing research where we adopted literature survey as a starting point in order extract a set of operations that is capable of at least describing state-of-the-art tools and proposals. However, questions concerning why such a framework is relevant and how it leverages descriptions and representations of exploration tasks, as well as reuse of exploration patterns are still lacking further explanations.

Since case studies have been considered a valuable tool for explanatory researches [27], we elaborate in this work a baseline strategy grounded on case studies over which we draw explanations addressing the value of a framework of exploration operations.

3 Conceptual View of an Exploration Framework

The lack of formalization of the exploration operations and the compositions that can be formed is the cause of relevant expressivity limitations, which can hinder many user strategies to solve complex information problems [17]. In Information Exploration, by expressivity we mean the ability of the computational system to support the execution of the user's problem solving strategy, which can be driven both by information scent, as in the case of Information Foraging Theory [24], or by some degree of curiosity, as normally occurs in Exploratory Search [31]. As a consequence, we can say that the more expressive a tool is, the larger the number of strategies it supports.

An example of user's problem solving strategy is the one proposed in [8]:

> "A person may not be able to specify the title or author of a book she is looking for, but may remember its approximate shelf location. In order to recognize the item, this person might go to this location and scan the shelves."

Using an exploration system, the person in the example above could start by querying books in nearby locations and scanning the abstracts in order to find the desired book. A different strategy could be, instead of issuing a query, to use the facet "location" in order to refine the current set of books. Although previous works have already called attention to the importance of identifying information seeking strategies [6–8] none of them proposed a precise and formal way of assessing tool expressivity.

Having a good understanding of the dimension of operations, parameters and results, and the dimension of the compositions that can be formed can benefit the Information Exploration field with proper instruments both to assess the expressivity of exploration systems and to leverage the system design process. This section presents an initial proposal of a framework of exploration operations that is useful for representing exploration solutions, comparing tools, and promoting reuse.

3.1 Preliminary Data Model

Since (semi) structured data published on the web shares a relational nature with traditional database systems, we describe our dataset in terms of a generic set-based model containing sets, items and ordered pairs representing binary relations between items.

A dataset D is formed by a set of items I, a set of Literals L, and ordered pairs $R: D \times D$:

$$D = I \cup L \cup R$$

There is a specific relationship $rtype: R \times L$ that maps all binary relations to their respective types, described by a literal in L. As an example consider the following relation:

rtype(<:Albert_Einstein, :Nobel_Prize_in_Physiscs>) → {:award},
where, <:Albert_Einstein, :Nobel_Prize_in_Physiscs> is a binary relation between two items.

In this work we adopt a simplified notation for describing restricted images of binary relations. Supposing a relation set of ordered pairs formed by people and their awards Award, the restricted image of the relation on the item :Albert_Einstein is denoted by:

Award = {<:Albert Einstein, :Copley Medal>,<:Albert Einstein, :Nobel Prize in Physics>},

$\text{Image}_{\text{:Albert\_Einstein}}(\text{Award})$ = {:Nobel_Prize_in_Physics, :Copley_Medal},

which is equivalent to:

:award(:Albert_Einstein*)* = {:Nobel_Prize_in_Physics, :Copley_Medal}

3.2 A Model of Exploration Operations

A common problem in the evaluation of exploration systems is to assess the degree of expressivity of such tools. At the time of this research, there isn't a common framework of operations that allows designers to identify neither which processing can be applied to a dataset nor in which sequence. The benefits of such framework are not restricted only to evaluation purposes but it also can leverage accurate representation of exploration tasks and reuse, as we demonstrate in the case study. However, in the same way as occurs with taxonomies of tactics and strategies, the completeness of such a framework is very hard to assess. In order to ensure a satisfactory coverage of the framework, we analyzed state-of-the-art tools and devised a set of operations that describes these systems, abstracting interface aspects. The main reference tools for the construction of the framework are: Tabulator [10], gfacet [19], /facet [20], Sewelis [16, 17], Visor [26], Rhizomer [18], MusicPinta [15], parallel faceted browser [13], Explorator [4] and its follow up RExplorator [14], and SeCo [12]. Therefore, the framework we present in this section is an ongoing work and it is not intended to present a final set of operations. Space reasons prevent us from showing the resulting model for all these frameworks, but we show one in section 4.3.

We describe below a first approximation of a set of operations comprehensive enough to at least describe the state-of-the-art exploration tools currently proposed in the literature, with regards to data manipulation:

- *Query*(items, queryPattern, matchingFunction): retrieves a set of items that matches the query pattern with the specified matching function;
- *Pivot*(items, relations): maps a set of items onto another set of related items;
- *Refine*(items, filter): restricts the current set of items to a set of items that matches the restrictions imposed by the explorer through the *filter* parameter;
- *GroupBy*(items, relation): groups a set of items based on a relation, which can be defined either by the data model or by the user as a computed relation. It is important to observe that this is an abstract definition, which can be specialized, for example, in a clustering operation, where the relation is a distance function;
- *FindPath*(sourceItems, targetItems): finds a set of relations that connects the two sets of items;
- *Rank*(items, fScore): ranks a set of items given a score function;
- *Eval*(functionalComposition, bindings): evaluates a functional composition against a set of parameter bindings. The *Eval* function is explained in detail in the next session.

We also define a syntax to describe the step-by-step application of the operations on sets of items. Each operation requires a block of actions that defines part of the behavior of the operation. The syntax for denoting operation applications is described below:

<ScopeSet>.<OpIdentifier>{|<ArgName>| <Body> }

<ScopeSet> is the set over which the operation is applied. The set can be a previously defined set or a result set of previous operations. Result sets are denoted by S_x, where x is a numerical index usually associated with the step number in which it was generated.

The application of an operation requires functional blocks of specifications delimited by braces ({}). <ArgName> stands for arguments passed to the blocks. These arguments are the items of the scope set passed one by one to be processed by the block. Items can be either simple items or relations, where the items in the relations are denoted as pairs inside parenthesis ((:itemA, :itemB)). <Body> stands for statements, such as, filtering predicates, and function applications, such as, query matching functions, other exploration operations, or functions defined by the user.

<OpIdentifier> is the identifier of the operation that will be applied to the scope set, such as *Rank* and *GroupBy*. The exceptions are the operations *Refine*, *Pivot*, and *Query* that can be identified through the block body, where we omit the operation identifiers. The *Refine* operation is identified by the presence of predicates in the <Body> part of the block, such as the filter of publications between 2001 and 2002 below:

S_3.{|p| 2001 ≤ :publicationYear(p) ≤ 2002}

The *Pivot* action is identified by the evaluation of a restricted image of a relation within the block:

S_1 ← Sc.{|p| :award(p)},

where, :award is the relation, Sc is a set of scientists, and S_1 is the resulting set of awards.

The *Query* operation is defined by the presence of a matching function within the body of the block, such as the second step of the case study presented in the next section:

S_2 ← S_1.{|ipc| keywordMatch(ipc, {{“semiconductor”, “silicon”, “led”, “insulator”, “transistor”}}),

where, keywordMatch matches an item passed as argument, denoted by “ipc”, with the keywords passed as parameters. The keywordMatch may use different string matching functions.

4 Exploration Case: Discovering Technological Trends

The goal of the case study presented here is to provide answers to why and how the proposed unified framework of exploration operations can benefit Information Exploration research. Therefore, we selected a common and documented exploration problem in patent analysis and used the framework to describe the exploration process and demonstrate a possible case of pattern sharing.

Patent datasets can be used as a source of information about changes in technological trends either in knowledge fields or in a company R&D strategy. Such information is valuable for the development of competitive intelligence of a company [22, 28]. The following task, raised in [28], has as its main goal to generate a report on technological trends for either a specific company or a patent classification domain. In order to demonstrate the expressivity of our framework in a complex task, we took a simplified version of the task presented in [28], which presents a system that allows patent analysts to trace changes in the activities in technology fields by analyzing patenting activities on these fields in two different time periods.

The changes in the technological landscape that can be identified by analyzing published patents in different time periods are observed by answering four main questions:

- Which industry fields have increased the level of attention throughout given periods?
- Which industry fields have decreased the level of attention throughout given periods?
- Which industry fields started to be addressed throughout given periods?
- Which industry fields stopped to be addressed throughout given periods?

The industry fields are mapped to the patent classifications in the International Patent Classification (IPC) system[1], which organizes a set of patent categories hierarchically. The level of attention of each IPC classification is measured by indicators that consider the age of the patents, the number of citations, the originality and generality of the patents, and the average age of the cited patents. For more details about the indicators, refer to [28]. For illustration purposes, let $l: P \rightarrow \mathbb{Q}$ be a function that maps a set of patent documents P into a numeric value in $\mathbb{Q}$ that represents the level of attention that the set is receiving.

The first step is to find the set of IPC classes related to some knowledge area:

1. $S_1 \leftarrow$ P.{|p| :hasIpc(p)}
2. $S_2 \leftarrow S_1$.{|ipc| *keywordMatch*(ipc, {{“semiconductor”, “silicon”, “led”, “insulator”, “transistor”}})

[1] http://web2.wipo.int/ipcpub/#refresh=page

The actions above are an attempt to find all classes related to the field of semiconductors by pivoting from the set of patents P to the set of IPCs through the *hasIpc* relation (step 1) and issuing a disjunctive keyword query on the set of IPCs with keywords related to the field of interest (step 2). Next, the explorer splits the set of patents into two sets published in different periods by, first, filtering out patents whose IPCs are not in the set of IPCs related with the field of interest using an intersection between the sets (step 3), and then, filtering patents published in the periods of interest (steps 4 and 5):

3. $S_3 \leftarrow S_1.\{|p| \cap(:\text{hasIpc}(p), S_2) \neq \{\}\ \}$
4. $S_4 \leftarrow S_3.\{|\ p|\ 2001 \leq :\text{publicationYear}(p) \leq 2002\}$
5. $S_5 \leftarrow S_3.\{|\ p|\ 2003 \leq :\text{publicationYear}(p) \leq 2004\}$

The goal of the next steps is to reorganize the data to answer the questions based on the levels of attention of each IPC:

6. $S_6 \leftarrow S_4.\mathit{GroupBy}\{|p|\ :\text{hasIpc}(p)\}$
7. $S_7 \leftarrow S_5.\mathit{GroupBy}\{|p|\ :\text{hasIpc}(p)\}$
8. $S_8 \leftarrow S_6.\{|(\text{id, ipc, patSet})|\ (\text{ipc}, \mathit{l}(\text{patSet}))\}$
9. $S_9 \leftarrow S_7.\{|(\text{id, ipc, patSet})|\ (\text{ipc}, \mathit{l}(\text{patSet}))\}$
10. $S_{10} \leftarrow S_4 - S_5$
11. $S_{11} \leftarrow S_5 - S_4$
12. $S_{12} \leftarrow S_8.\{|(\text{ipc, attentionLevel})|\ \text{attentionLevel} < S_9(\text{ipc})\}$
13. $S_{13} \leftarrow S_8.\{|(\text{ipc, attentionLevel})|\ \text{attentionLevel} > S_9(\text{ipc})\}$
14. $S_{14} \leftarrow S_8.\{|(\text{ipc, attentionLevel})|\ \text{attentionLevel} = S_9(\text{ipc})\}$

Steps 6 and 7 groups the sets of patents published within the two different periods by their IPCs. Having the groups of patents per IPC, the explorer applies the function $l: P \rightarrow Q$ to extract the level of attention for each IPC (steps 8 and 9). Next, the explorer splits the set of all IPC classifications into classifications that started to gain attention along the periods (step 10), classifications that are no longer addressed from one period to the next (step 11), classifications that have increased the level of attention along the periods, measured by the function l in steps 8 and 9 (step 12), classifications that have decreased the level of attention along the periods (step 13), and classifications that remained with the same level of attention.

For a set of ordered pairs denoting a relation, we can obtain restricted images using the name of the set and parenthesis, as in steps 12 through 14: $S_9(\text{ipc})$, where S_9 is a relation between IPCs and levels of attention, *ipc* is the domain element and the returned image is the level of attention for that *ipc*. For example, suppose the relation between two IPC identifiers and their respective levels of attention E =

{<:A61H_33/02, 12.5>, <:F21Y_101/02, 20.4>}. The syntax to obtain the image of E restricted on A61H_33/02 is E(:A61H_33/02) = {12.5}.

4.1 Generalizing and Reusing Exploration Patterns

The sequence of steps to solve the problem of tracing changes in the technological landscape can be generalized both in the dimension of a related domain of problem and in the dimension of related tasks within the same domain. As an example of a generalization within the same domain, imagine that the same task needs to be executed but the analyst should now analyze the changes in two more recent periods: from 2010 to 2012 and from 2013 to 2014. Therefore, the expression needs to be re-evaluated for two different ranges of years:

$$Eval(1..14, \text{step4.2001\%2010}, \text{step4.2002\%2012}, \text{step5.2003\%2013}, \\ \text{step5.2004\%2014})$$

The Eval function receives as input the expression of the composition that will be reused and re-evaluated, identified by a range of step indices (1..14), and the argument replacements defined by the identifier of the step (step1, step4, step5, …), the argument that will be replaced followed by the replacement symbol "%", and the new argument for the evaluation (step4.2001%2010). Notice that not only literals can be replaced, but also variables by functional expressions, e.g., ipc%avg(ipc, 2003-2013).

The second possibility is to reuse solutions in different related domains. For example, the reuse of the solution from the patent analysis domain to the related research papers investigation domain can also be carried out by parameterizing the expression. Consider a set of research papers Rp and a relation hasTopic: Rp X T, where T is a set of research topics. The re-evaluation is carried out as follows:

$$Eval(1..14, \text{step1.P\%Rp}, \text{step1.:hasIpc\%:hasTopic}, \text{step6.:hasIpc\%:hasTopic}, \\ \text{step7.:hasIpc\%:hasTopic})$$

In the re-evaluation above, the argument replacements are the set of patents P by the set of papers Rp, and the relation hasIpc in the steps 1, 6, and 7 by the relation hasTopic.

4.2 Comparing Exploration Tools

Once the case study is grounded on an expressive exploration language we can also analyze the adequacy of exploration tools for solving similar tasks and devise insights with regards to how expressive a tool is for solving a class of problems. For demonstration purposes, we selected the tools Cambria Patent Lens[2] and gfacet [19].

Solving the task presented in this case study is quite complex in Patent Lens due to the lack of the exploration primitives *Pivot*, *Group, Map, Intersect*, and *Diff.* As a

[2] https://www.lens.org/lens/collection/5116

result, it disallows operations targeting the IPCs, such as the keyword search in step 2, which forces the explorer to interrupt the task, pick another tool, find and annotate the IPCs of interest, return to Patent Lens, and filter the patents by the manually annotated IPCs. Patent Lens does not allow grouping a set of patents by some criteria and map collections of items onto other collections, making steps 6 to 9 impossible to achieve. Since set operations are not offered also, steps 10 and 11 are not allowed.

Differently from Patent Lens, gfacet features the *Pivot* action, which allows the explorer to carry out exploratory operations over multiple types of items, thus, making steps 1 and 2 possible. However gfacet still lacks *Group*, *Map*, and set based operations. Moreover, there is a shortcoming in the expressivity of the *Refine* operation where it is not possible to apply restrictions on a range of values, such as the restrictions on the publication year of the patents on steps 4 and 5.

As a conclusion, we can observe that Patent lens is not a feasible tool for exploration tasks that requires analysis of multiple and related types of items, such as patents and IPCs, due to the lack of the *Pivot* primitive. Even though gfacet does provide a *Pivot* primitive, gfacet lacks expressivity for tasks that require refinements through ranges of values. Furthermore, tasks requiring analysis and insights computed by operations and measures applied to sets of resources are not well supported by both Patent Lens and gfacet.

4.3 Expressivity Analysis of gfacet

The concept of expressivity of an exploration tools is related both to the range of questions that can be answered, and to the complexity of those questions [16]. Although expressivity is a fundamental parameter for tools' design, it is usually neglected in scientific publications of new exploration tools, which typically carry out user studies at the interface level. Nothing is reported with regards to the support to classes of exploration tasks or the level of support to the exploration process considering, for example, whether alternative combinations of actions are possible or not. In this section we demonstrate how the framework of operations serves as a tool to carry out a qualitative analysis of the support to the exploration process. In order to demonstrate how the analysis can be carried out using the framework we selected the exploration tool gfacet.

We analyze the expressivity of gfacet in two dimensions: The first dimension concerns the set of primitives and the expressivity of each single operation. The second dimension concerns the functional compositions that can be elaborated from the successive application of the primitive operations.

Starting from the set of primitive operations, by analyzing gfacet's interface and research paper we obtain the following list of operations, where “keywords”, “relation”, “relation$_x$”, and “object$_x$”, are variables whose values are defined by the user, *D* represents the dataset, and *S* represents some partial result set:

- Keyword query:
  D.{|item| match(item, :subject(?x) = item)}.{|item| keyword-match(item, keywords)}

- Pivot:

  $S.\{|\text{item}|\ \text{relation(item)}\}$, where, $\text{length(relation)} = 1$
- Refine (property filters):

  $D.\{|\text{item}|\ \text{relation}_1\text{(item)} = \text{object}_1 \wedge \ldots \wedge \text{relation}_n\text{(item)} = \text{object}_m\}$
- Refine (keyword filters)

  $S.\{|\text{item}|\ \text{:label(item) INCLUDES keywords}\}$

The starting point is a keyword query. The first limitation is the impossibility of matching the keywords on any type of item. The keyword query is restricted to items that are subjects of other items, related by the dcterms:subject[3] property (:subject). Another limitation is the impossibility of expressing disjoint keyword expressions, all keywords are always connected by logical ANDs.

The *Pivot* operation in gfacet is carried out by choosing one of the relations of the items in some set S. One limitation of *Pivot* is that the relation should be a single relation (length(relation) = 1), which disallows the possibility of expressing property paths. For example, consider a scenario where the user needs to pivot from a set of inventors to the set of years they were born, and the relation is formed by the property path ":birthdate:year". The user will have to pivot twice instead of entering the property path of interest directly. Although this is not a big problem in this case, in cases where items are connected by more complex property paths, such as the one connecting Politicians and States presented in [3]. Once the property path is discovered, it is desirable to be able to refer to it in future pivoting as a single property.

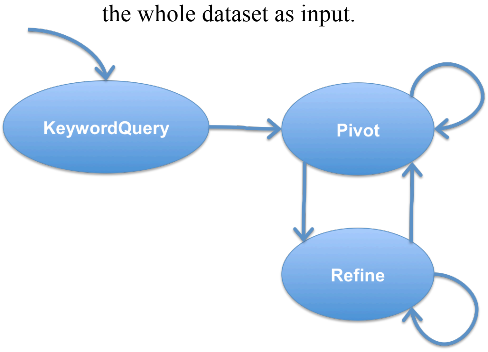
The gfacet comes with two possibilities of refinement: refinement through property filters and refinement by keyword filters. Using property filters the user can filter sets of items, representing RDF subjects, by selecting a filter composed by a property and an object ($\text{relation}_x\text{(item)} = \text{object}_x$). A set of items *S* can be filtered by the conjunction of any set of filters. The other possibility of refinement is through typing keywords in a text box to keep only items whose label contains the keywords. The limitations of the gfacet's *Refine* operation are due to the filtering expressions, as it is not possible to apply disjunctions of filters.

In order to assess the range of functional compositions that can be formed, we analyze what are the possible sequences of operations allowed and how the results of previous operations can serve as input for the next operations. The diagram in Fig. 1 shows the compositions that can be formed, where the arrow means "the output of the operation at the source can be used as input for the operation at the target" and the arrow with no source operation means that the entire dataset serves as input for the operation and it is also a starting point for the exploration. For demonstration purposes, we considered only the refinement through property filters, which is one of the main features of gfacet.

Fig. 1. Diagram representing how the operations can be composed. The arrows indicate that the output of an operation (arrow source) can be used as input to another (arrow target). The

[3] http://dublincore.org/documents/dcmi-terms/#terms-subject

arrow with no source operation indicates a starting point for the exploration and the usage of the whole dataset as input.



From Fig. 1 we can observe that *KeywordQuery* is the starting point of the exploration and it cannot be applied over the results of previous actions. *Pivot* and *Refine* can be applied over the results of *KeywordQuery* and the explorer can also pivot both from the results of a previous pivoting and from the results of a previous refinement. The *Refine* operation follows the same dynamics of *Pivot*.

From this model we can draw some conclusions. First, the starting point is always a known item search tactic expressed as a keyword query over DBpedia subjects, which, can be a problem for users that are not able to express their needs through keywords due to lack of knowledge. Second, the exploration can be composed of many sequences of pivoting and refinements, starting from a keyword query, but, there isn't an operation that combines results from different sequences (e.g. Union), and it is not possible for the sequences to converge at some point. Therefore, the paths can assume only the format of a tree and it is not possible to process items from different leaves for comparison purposes, for example.

5 Related Works

Many works on visualization research have pointed out the necessity of filing the gaps between visual data representations and information-seeking tasks [1, 2, 25, 29], which generated taxonomies and ontologies addressing at least these two dimensions.

The work in [29] organizes information objects within seven types, ranging from 1-dimensional objects to trees and networks, and common tasks carried out on those objects in a taxonomy of object types by task, aiming at facilitating further discussions on the topic.

The work in [2] also emphasizes the need of approximating the design and evaluation of visualization systems to the users' tasks. It examines how the limitations in information visualization systems are originated from analytic gaps between current systems and higher-level analysis tasks through case studies. The analysis generated a taxonomy of analytic tasks, presented in [1], which comprises many exploration strategies.

A more recent approach [25] aims at the collaborative construction of a visualization ontology using Semantic Web technology. The VISO ontology contains seven modules with the goal of characterizing, among other aspects, graphic attributes (GRAPHIC), data and data structure (DATA), tasks and actions (ACTIVITY), and facts presented in the literature with regards to visualizations (FACTS). Although the proposal of the ACTIVITY module seems to overlap with our proposal, in practice, it is mostly concerned with visualization operations, such as, semantic and geometric zooms[4].

In general, the works on a common vocabulary of actions in the visualization field addresses the problem of which tactics the visualization should aid and how the actions relate to certain data types, but, frequently such taxonomies lack details on how these actions actually translate to data processing operations. Some questions remain open, such as, which parameters/options are required or possible? What are the results of an action? How can the actions be combined into more complex actions? Is it possible to employ alternative sequences of actions to achieve a solution?

In summary, these vocabularies are useful for an abstract conceptualization of what the user can do when exploring a dataset and can be naturally related to the framework we are proposing. For example, Marchionini's "known item search" and "navigation" tactics [21] characterize *Query* and *Pivot* operations. Shneiderman's "filter" and "relate" tasks [29] can also characterize the *Refine* and the *FindPath* operations. Nevertheless, there is still a gap between these vocabularies and the accurate representation of an exploration task, which hinders generalization and reuse of paths and solutions.

6 Conclusion and Future Directions

This work, a follow up of the work in [23], presents a case study on a framework of exploration operations demonstrating how it can be used to describe exploration solutions, analyze the adequacy of exploration tools with regards to a specific task, and promote generalization and sharing of exploration solutions.

Through the case study we illustrate the relevance of a unified and expressive framework of exploration primitives to advance the discussion on the design and comparison of Information Exploration tools. We also argue the benefits that can be achieved by modeling exploration solutions as compositions of operations, bringing the possibility of generalization and reuse of exploration patterns through parameterization of the functional compositions. We selected the field of patent analyses due to the considerable complexity and recurrence of some exploration tasks among the community of patent analysts.

As ongoing work, we are finalizing the formal description of the framework and implementing a proof of concept tool to carry out user studies in order to also carry out a user-centered evaluation of our approach.

[4] https://github.com/viso-ontology/viso-ontology

Acknowledgement

The authors were partially supported by CNPq project 557128/200-9 National Science, Technology Institute on Web Science, CAPES, and Google Research Program.

7 References

1. Amar, R. et al.: Low-level components of analytic activity in information visualization. Proc. - IEEE Symp. Inf. Vis. INFO VIS. 111–117 (2005).
2. Amar, R. a., Stasko, J.T.: Knowledge precepts for design and evaluation of information visualizations. IEEE Trans. Vis. Comput. Graph. 11, 4, 432–442 (2005).
3. Araujo, S. et al.: Fusion - Visually Exploring and Eliciting Relationships in Linked Data. Proceedings of the 9th International Semantic Web Conference on The Semantic Web - Volume Part I. pp. 1–15 Springer-Verlag, Berlin, Heidelberg (2010).
4. Araújo, S. De, Schwabe, D.: Explorator: a tool for exploring RDF data through direct manipulation. Linked data web. (2009).
5. Baldonado, M.Q.W., Winograd, T.: SenseMaker: An information-exploration interface supporting the contextual evolution of a user's interests. Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '97. pp. 11–18 ACM Press, New York, New York, USA (1997).
6. Bates, M.J.: Information search tactics. J. Am. Soc. Inf. Sci. 30, 4, 205–214 (1979).
7. Bates, M.J.: The design of browsing and berrypicking techniques for the online search interface. Online Inf. Rev. 13, 5, 407–424 (1989).
8. Belkin, N.: Cases, Scripts, and Information-Seeking Strategies: On the Design of Interactive Information Retrieval Systems. Expert Syst. Appl. 9, 3, 379–395 (1995).
9. Berners-lee, T. et al.: Tabulator : Exploring and Analyzing linked data on the Semantic Web. Methodology. 2006, i, 6 (2006).
10. Berners-lee, T. et al.: Tabulator: Exploring and Analyzing linked data on the Semantic Web. Swui. 2006, i, 16 (2006).
11. Bizer, C. et al.: Linked Data - The Story So Far. Int. J. Semant. Web Inf. Syst. 5, 3, 1–22 (2009).
12. Bozzon, A. et al.: Exploratory search framework for Web data sources. VLDB J. 22, 5, 641–663 (2013).
13. Buschbeck, S. et al.: Parallel faceted browsing. CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13. p. 3023 ACM Press, New York, New York, USA (2013).
14. Cohen, M., Schwabe, D.: Support for Reusable Explorations of Linked Data in the Semantic Web. In: Harth, Andreas and Koch, N. (ed.) Current Trends in Web Engineering. pp. 119–126 Springer Berlin Heidelberg (2012).
15. Dimitrova, V. et al.: Exploring exploratory search. Proceedings of the 2nd International Workshop on Intelligent Exploration of Semantic Data - IESD '13. pp. 1–8 ACM Press, New York, New York, USA (2013).
16. Ferré, S.: Expressive and Scalable Query-Based Faceted Search over SPARQL Endpoints. Springer International Publishing, Riva del Garda (2014).
17. Ferré, S., Hermann, a.: Reconciling faceted search and query languages for the Semantic Web. Int. J. Metadata, Semant. Ontol. 7, 1, 37 (2012).
18. García, R. et al.: Rhizomer : Overview, Facets and Pivoting for Semantic Data Exploration. Int. Work. Intell. Explor. Semant. Data. (2013).
19. Heim, P. et al.: gFacet : A Browser for the Web of Data. C, 2005–2008 (2008).

20. Hildebrand, M. et al.: /facet: A Browser for Heterogeneous Semantic Web Repositories Michiel. 272–285 (2006).
21. Marchionini, G.: From finding to understanding. Commun. ACM. 49, 4, 41–46 (2006).
22. Mukherjea, S. et al.: Information retrieval and knowledge discovery utilizing a biomedical patent semantic Web. IEEE Trans. Knowl. Data Eng. 17, 8, 2005–2008 (2005).
23. Nunes, T., Schwabe, D.: Exploration of Semi-Structured Data Sources. 3rd Int. Work. Intell. Explor. Semant. Data (IESD 2014). (2014).
24. Pirolli, P.L.T.: Information Foraging Theory: Adaptive Interaction with Information. Oxford University Press, Inc., New York, NY, USA (2007).
25. Polowinski, J., Voigt, M.: VISO: A Shared, Formal Knowledge Base as a Foundation for Semi-automatic InfoVis Systems. CHI ’13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13. p. 1791 ACM Press, New York, New York, USA (2013).
26. Popov, I. et al.: Connecting the dots: a multi-pivot approach to data exploration. Semant. Web–ISWC …. 7301, 553–568 (2011).
27. Rowley, J.: Using case studies in research. Manag. Res. News. 25, 1, 16–27 (2002).
28. Shih, M.J. et al.: Discovering competitive intelligence by mining changes in patent trends. Expert Syst. Appl. 37, 4, 2882–2890 (2010).
29. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. Proceedings 1996 IEEE Symposium on Visual Languages. pp. 336–343 IEEE Comput. Soc. Press (1996).
30. White, R.W. et al.: Exploratory search interfaces. ACM SIGIR Forum. 39, 2, 52 (2005).
31. White, R.W., Roth, R.A.: Exploratory Search: Beyond the Query-Response Paradigm. Synth. Lect. Inf. Concepts, Retrieval, Serv. 1, 1, 1–98 (2009).
32. Wildemuth, B.M., Freund, L.: Assigning search tasks designed to elicit exploratory search behaviors. Proc. Symp. Human-Computer Interact. Inf. Retr. - HCIR ’12. C, 1–10 (2012).