# VIQS: Visual Interactive Exploration of Query Semantics

**Christina Christodoulakis**
University of Toronto
christina@cs.toronto.edu

**Eser Kandogan**
IBM Research, Almaden
eser@us.ibm.com

**Ignacio G Terrizzano**
IBM Research, Almaden
igterriz@us.ibm.com

**Renée J Miller**
University of Toronto
miller@cs.toronto.edu

## ABSTRACT

Analytics platforms such as IBM's Watson Analytics™ are collecting metadata about their use, including user queries on uploaded datasets. The analysis of this metadata may be valuable in improving services, such as query recommendation and automatic data visualization. However, analysis of metadata is difficult not only in terms of scale but also in terms of complexity. Generalizing and exploring query patterns across users and datasets is challenging. Abstractions are likely to help bridge differences in specifics (e.g., column names and query details), particularly in semantics. For example, a single query, *"What is the trend of sales over year?"* could be abstracted in many different ways (e.g., *"What is the trend of financial gain over time?"*). In this paper, we describe our process of creating a dataset of query semantics, starting from initial metadata extraction from query logs to semantic expansion using WordNet. To help system designers effectively browse and understand patterns of use, we developed VIQS (Visual Interactive Query Semantics), a system that extracts query semantics from query logs over multiple datasets, and allows users to explore underlying patterns visually. We present results from an informal interview study along with specific insights regarding popular query patterns from 3-months of data. We believe the analytic process, as well as the specific insights on query patterns, will benefit the design of analytics platforms.

## CCS Concepts

•**Human-centered computing** → **Systems and tools for interaction design;** *Information visualization;*

## Author Keywords

Information Seeking & Search; User Interface Design

## INTRODUCTION

Several initiatives are emerging to support the publication, sharing, and analysis of data, including in open data platforms such as data.gov [8] and GenBank [3]. Open science initiatives, including LabBook, aim to accelerate scientific discovery by making experimental data and lab notes accessible to the broader research community [14]. In business, open analytics platforms such as IBM Watson Analytics™ allow users to share data, models, processes, and insight [26].

Large-scale use of open analytics platforms generates tremendous amounts of metadata on users, datasets, queries, and visualizations. Such metadata is of particular interest to system designers as a resource to understand their users and further improve usability and functionality provided. For example, IBM Watson Analytics provides users with an initial set of query and visualization recommendations upon uploading their datasets. Query and visualization recommendations could be significantly improved by tracking which queries users picked among a recommended set of queries and which visualizations were most useful.

While the promise of using metadata to understand data analysis is enticing, the reality is that it is very challenging. There are several reasons. (1) In open analytic platforms, the user base is quite diverse with different interests and analysis patterns. (2) Datasets contributed by users have very diverse schemas (e.g., column names). Such diversity makes it challenging to derive high-level patterns of use across users and datasets. Raw metadata and queries need to be enriched with semantics. (3) The addition of semantics increases the size and complexity of the metadata.

We consider the analysis of query patterns across users and across datasets to inform the design of query recommendation systems. To assist system designers, we created semantically enriched query metadata by analyzing a very large query log dataset (from Watson Analytics) and we built a visual interactive tool (VIQS) to allow users to easily explore and understand query semantics. Our contributions include: (1) a novel approach to the abstraction of structured data-analysis queries using semantic annotations of schemas and (2) a visual analysis tool for analyzing logs of structured queries over a large repository of diverse users and datasets.

The rest of the paper is organized as follows. First, we cover related work in the areas of information management, semantic data, and visual analytics. Then, we describe our approach in extracting queries from logs and our semantic enrichment of column names as well as queries. We then present our visual interactive query semantics (VIQS) tool for analyzing this metadata. Next, we describe an informal study to assess initial feedback on our tool along with a use-case to describe some



*ESIDA'17,* March 13 2017, Limassol, Cyprus

 ISBN 978-1-4503-4903-1/17/03…$15.00

DOI: http://dx.doi.org/10.1145/3038462.3038470

of the insights found for popular query patterns. We conclude with a discussion of our approach.

## RELATED WORK

The field of visual analytics is deep and rich, but there is little work on the visual analysis of complex data analytic queries. While query recommendation is well studied, the visual analysis of queries is not. Similarly, while conceptual summarization is well studied, to the best of our knowledge, it has not been applied to structured queries or query logs over diverse datasets. We believe this work will inspire more use of interactive visualizations in understanding and summarizing queries and query patterns. Here, we focus on applying insights we derive to inform query recommendation.

### Structured Query Summarization

Identifying queries that users are most interested in is critical for optimization of system performance and support of exploratory data analysis. In structured databases (including data warehouses), a number of query recommendation systems have been proposed [1, 2, 11] to help users form queries or find similar or related data. When queries are structured (including multiple tables, join and selection conditions, projections, and possible groupings and aggregations), the form of the query can be used to define query similarity [2, 5, 12, 21, 30, 31] and then be utilized to find similar sets of tables and columns. Query similarity can also be used to identify important queries [29]. In addition to using a syntactic notion of query similarity, some of these approaches also measure how related the query results of the recommended query would be to a given query [9, 23]. In general, a combination of syntactic query similarity and instance-based similarity can be useful for query recommendation over a single (coherent) database where all users are using the same schema.

A differentiating factor with our work is the very large number of heterogeneous tables (potentially from different domains) that use different vocabularies and naming conventions for their table column names. For example, even when column names match, they may carry different semantics. As such defining query similarity by relying on the use of common names for columns, or by relying on the contents of this heterogeneous collection of data is not feasible. Furthermore, these database techniques do not provide a way of recommending queries over new tables not already in the query log. In our work, we explore how to map column names to semantic concepts and we use a concept hierarchy to group and summarize queries.

### Semantic Query Summarization

Semantic annotations have been used in query summarization previously, but mostly in document search where queries are phrases or bags of words, rather than structured queries [16, 28]. Query clustering is a process used to discover frequently asked questions or most popular topics on a search engine. Query clustering uses either query contents (keywords) or document clicks (the user accepted a result as related to her query), or a combination of the above. In such cases, when two queries contain the same or similar terms, or lead to the selection of the same or similar document they are considered similar. Use of semantics for query clustering is mainly based on synonyms (e.g., in WordNet [10], each concept will have a set of synonyms). Often these query summarization techniques do not use or take full advantage of the concept hierarchy (the hypernym relationship between concepts) [20]. In contrast, document summarization work (where the goal is summarizing the query results, i.e., the documents, rather than the queries) does use the concept hierarchy [7, 17, 22].

Because our queries are structured, not bags of words, we use both a mapping of column names to concepts (to abstract away from the choice of vocabulary used in different tables where different terms may be used for the same concept) and the concept hierarchy to allow for summarization of queries at multiple levels of abstraction. We used WordNet [10], but could alternatively (or in conjunction with WordNet) have used other ontologies such as YAGO [24].

### Visual Analysis

With some exceptions there has been little work in visualization of structured query logs [29], and none in the type of logs generated in analytic environments over data lakes. Guo et al. [13], explore logs of interactions with a visual analysis application to better understand how interactions lead to insight generation. There has been exceptional work done in document summarization [18, 25, 27]. TextArc [18] exposes frequency and distribution of words of a body of unstructured text with no markup or meta information in a large connected word cloud using brightness and size of words in the word cloud to convey frequency of use. In VIQS we convey frequency of use of both query templates and concepts that have been queried over, however we are also restricted in displaying the information in a manner that maintains the structure of a query. With Phrase Nets [25], Ham et al. generate visual overviews of unstructured text documents. They extract and visualize networks of terms from bodies of unstructured text. Word Tree [27] provides keyword-in-context views of a document enabling rapid querying and exploration of large documents. Structure of phrases is maintained using the branching layout of a tree diagram. In VIQS the structure of queries is a bit more complicated to display than that of a phrase, as it is essential to convey ordered sets and the positioning of the column concepts referenced. Docuburst [7] is a tool for summarizing and comparing document content (unstructured text) by combining word frequency with human created structure in lexical databases to create interactive semantic summaries of texts which are comparative at a glance. The authors leverage the hypernym structure among WordNet verbs and nouns to generate visualizations that allow exploration and comparison of document summaries at varying levels of semantic granularity. Docuburst compares to VIQS in that our goal is also to summarize information. However, VIQS addresses summarization of structured query logs (instead of text documents) over huge variety of data tables with varying schema. This variety of data and their schema require concept tagging and abstraction. In doing so, we can understand what analytic queries people may be interested in given a new dataset, and it can lead to better query recommendations and eventually guided data analysis and automatic report generation. With

VIQS, we visualize query logs using table column semantics and query structure, providing exploration of the semantic logs at varying layers of semantic granularity. We support interactive query analysis with semantic zoom, selectable focus on query templates or column semantics, and access to provenance (original queries).

## QUERY SEMANTICS EXTRACTION

We begin by describing how we extract query semantics metadata from logs of structured data-analysis queries. The logs contain queries issued over a large number of tables by a large number of users within the IBM Watson Analytics platform. Raw queries in logs refer to actual column names drawn from a large vocabulary, some fairly long containing actual sentences, such as survey questions. Inspection of these logs without any pre-processing yields no insights. Here, we describe how we process these logs and enhance them with semantics that permits meaningful aggregation of similar queries and meaningful navigation between related query abstractions.

### Watson Analytics Logs and Query Syntax

IBM Watson Analytics is a web-based data analytics platform for interactive analysis of datasets. We analyzed a three-month period of logs from which we extracted over 915,000 queries from tens of thousands of users. In Watson Analytics, users typically upload a CSV file, containing column names that are more readable than typical column names in a relational database as they are used in natural language queries. As such queries and visualizations are easier to understand and share. Users are presented with a set of possible visualizations based on a set of query templates.

Query templates are composed of keywords indicating a data analysis task, column names, and occasionally data values. While queries typically refer to at least two columns, some templates support more. At the time of our analysis, for queries referencing two columns in placeholders X and Y, we identified eight different templates (i.e., *"How do the values of X* ***compare by*** *Y?"*, *"'What is the* ***breakdown of*** *X by Y?"*, *"What is the* ***trend of*** *X over Y?"*, *"What is the* ***relationship between*** *X and Y?"*, *"What is the* ***contribution of*** *X over Y?"*, *"How does X* ***relate to*** *Y?"*, *"What is the* ***grouping of*** *X by Y?"*). Query templates can also contain data values to filter the data, (e.g., *"What is the breakdown of sales by region for* ***tablets****?"*). Finally, queries can contain specific keywords that suggest a particular aggregation (e.g., *average*, *maximum*, and *total*).

### Query Extraction from Logs

In addition to obtaining the query text, table, and targeted table columns, the query extraction process requires identifying the query template to which the query conforms. We determine the query template by comparing the raw query text with predefined regular expressions that correspond to the supported query templates.

Across different database tables, users typically use different words or variations of a word to describe fundamentally the same concepts. To understand query semantics, we use WordNet [10], a hierarchical lexical database with 155,287 words to tag columns with semantic concepts of varying levels of abstraction. Given a dataset with column named *per_anum* and another dataset with a column named *hourlyRate*, we can abstract both columns to the concept *time_period*. WordNet groups words that are synonymous into synsets, and different senses of a word are described in different synsets. Synsets are connected to each other with semantic relations, which for nouns are hypernyms, hyponyms, coordinate terms, meronyms and holonyms. To associate column names with semantics, we tokenize column names and apply standard lemmatization. Resulting tokens are then used to associate semantic tags with each column. For each tag we perform a search in WordNet. If a matching synonym set is found in the thesaurus we then proceed to connect that token to it. Each WordNet node representing a sense or a word is further connected to hypernym trees. This process is similar to topic expansion and has been used previously for query clustering as well [17].

Semantic annotation of column names in a table schema, without taking into account data type, column values, or context (i.e., surrounding column names in the table schema) can lead to errors. For example, a column named *state* could be mapped to multiple senses including *province*; *the group of people comprising the government of a sovereign state*; *the way something is with respect to its main columns*; or *state of matter*. We use the table context to narrow down the likely senses for an attribute. If surrounding column names are *street*, *zipCode* and *city*, the best sense is likely *province*. Upon processing our three-month log data, we identified nearly 4 million columns, which are eventually associated with 21,000 semantic concepts from WordNet at some level of abstraction.

### Generating Query Semantics

For recommendation purposes, our goal is to understand the popularity of queries. As we discussed however, computing counts of specific raw queries is not likely to result in a very meaningful generalization that could be applied to a new dataset for recommendation purposes, simply because the same column name is highly unlikely to be present in this new dataset. As such, in previous steps we performed semantic expansion of column names for the purpose of generalizing. For example, if a specific query is *"What is the breakdown of revenue by state?"*, we want to be able to generalize this query at several levels, such as *"What is the breakdown of financial_gain by administrative_region?"*, *"What is the breakdown of amount_money by location?"*, *"What is the breakdown of measure_quantitative by location?"*, etc. Each of these abstractions, we call a query semantic permutation, as it takes a semantic concept from all possible semantic expansions on the column. The totality of all these permutations is referred to as query semantics, in essence representing all possible abstractions at various levels.

What each query semantic permutation represents is an opportunity for another query from another user on another table to coincide in terms of its semantics. For example, a query on a table, such as *"What is the breakdown of sales by county?"* and another query on another table such as *"What is the breakdown of revenue by state?"* can all contribute to the evidence of a higher-level permutation such as *"What is the breakdown of financial_gain by administrative district?"*. Our argument

is that it is the evidence of these higher-level query semantic permutations that can potentially be very useful in making query recommendations on an existing or new table if columns of this table match the permutation.

Note that the number of permutations is large, as we take all possible semantic abstractions on one column in a query with all possible semantics abstractions for a second column. We only create permutations for which there are at least some threshold minimum number of queries in the log and our query templates have at most three columns (most have only two). We generated nearly 900,000 query semantic permutations from the three-month query log.

#### Semantic Graph Compaction

Following the process of semantic enrichment, we generate a large semantic graph of concepts, where the granularity of concepts range from tags generated during column name tokenization to the most abstract of concepts found in the hypernyms (parent concepts) of those tags. In tagging columns with hierarchies of concepts from WordNet, we noticed that each column could be overwhelmed with long hierarchies of semantic concepts. To create a clean semantic graph and eliminate redundant nodes, we follow the next two steps.

After a certain level of abstraction concepts became much too general, losing their value in successfully referencing a particular set of columns (e.g., all concepts related to nouns eventually converge to *abstract_entity* and *entity*). Upon closer evaluation it was determined that concepts above eight levels of abstraction from original column names were simply too abstract to have any value in query summarization.

We also decided to remove unnecessary concepts from our semantic graph given the following criteria. If a concept is connected to a number of columns beneath a minimum threshold, they are unlikely to be of importance.If a concept simply linked to a parent concept without any other outgoing links, the parent concept can just as easily act as a summarizing concept. On the other hand concepts that are parents of multiple concepts (hubs) are of very high importance as they function as a summary of multiple concepts and are preserved in the graph.

#### Query Semantic Metadata

The end result of the extraction process results in three data sets that will be used by VIQS: (1) Column semantics; (2) Query semantic permutations; and (3) Query templates.

The column semantics dataset contains a JSON object per concept with: (1) an identifier for each semantic concept, (2) a label describing the concept, (3) a column count measuring how many columns are mapped to this semantic concept, (4) the abstraction level measuring the shortest distance of the semantic concept from a column, (5) a set of semantic concepts related through a hypernym relationship which for simplicity we call *isA*, (6) a set of semantic concepts related through a hyponym relationship, which we label as derivedFrom, and (7) query statistics, providing counts of queries where this semantic concept occurred at placeholder i of query template with j placeholders (e.g., *compare_2_1* records occurrence of a concept in the first placeholder in a *compare* template with 2 placeholders). The JSON object recording metadata regarding column semantics of the concept *financial_gain*, recorded that this concept is at abstraction level 3, mapped to 58,808 columns across all datasets, isA *sum_of_money*, and is derived from semantic concepts *income* and *profit*, has occurred in the *compare* template with two columns (i.e., compare X by Y), used 14,388 times in placeholder X and 3,133 times in placeholder Y. This dataset contains a graph structure through *isA* and *derivedFrom* attributes, as well as a set structure with query counts per template and placeholder.

The metadata about query semantic permutations contains: (1) a specific query semantic permutation (i.e., a tuple of query template and specific semantic concepts at each column placeholder), and (2) query count, measuring how many queries are mapped to this permutation. A JSON object describing the query semantic permutation metadata for query template *trend_2* (i.e., two column trend query) for semantic concepts X=*monetary_unit* and Y=*quarter* reflects that 135 queries match exactly this specific permutation. This dataset contains several JSON attributes (template, columns, queryCount, etc.), along with an ordered set of semantic concept references.

Lastly, metadata about query templates contains: (1) template ID, (2) template syntax, and (3) query count, measuring number of queries with this template. The JSON object recording metadata for the *compare_2* template records its template syntax and the fact that 223,331 queries matched this template.

#### VIQS

Previous query recommendation approaches focused on a single database or data warehouse with a single known data schema. Watson Analytics on the other hand has structured queries, expressed in pseudo-natural language, over data tables from several domains by multiple users. Understanding query logs produced by such systems is informative to recommendation system designers to understand popular query patterns in data.

The overall promise is that if the system has knowledge of query patterns over say survey data, such patterns can be used in recommendation systems via rules or can be learned from to build recommendation models, such that a new user uploading new survey data can be guided through the process via sample queries. For example, a query on a customer satisfaction dataset such as *"How do the values of customer satisfaction compare by flight time?"*, can be leveraged to recommend to another user a query such as *"How do the values of job satisfaction compare by age?"* on a human resources survey dataset, if there is sufficient popularity of a query pattern such as *"How do the values of emotional_state compare by measure_quantity?"*, or even more specifically *"How do the values of satisfaction compare by time?"*. To build such a knowledge base, we built VIQS to help system designers explore and understand query patterns. VIQS enables exploration of query logs at varying levels of granularity, from very abstract concepts to the original queries themselves.

After interviewing system designers we identified that they need support in exploring (1) overall popularity of query templates (2) overall popularity of semantic concepts (3) popular-

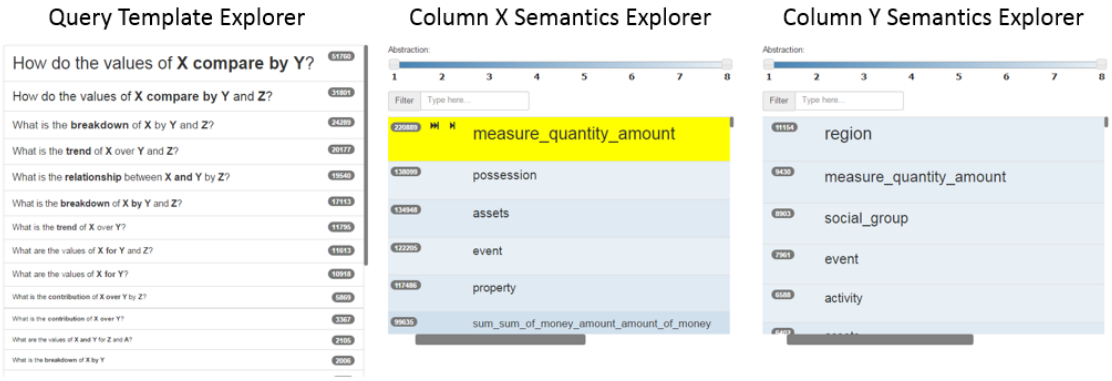


**Figure 1. The Pattern Browser. Users can select templates and assign a semantic concept from various levels of abstraction to build a query semantic they are interested in exploring.**

ity of semantic concepts by query templates, (4) popularity of query templates by semantic concepts (5) overall popularity of combinations of semantic concepts, (6) popularity of combinations of semantic concepts by query templates and vice versa. For example, what are common queries over particular concepts (e.g., *income*), what concepts do people explore in "trend of" queries (e.g., *financial gain*), what concepts people seek trends over (e.g., *time*), what attribute combinations occur more (e.g., *revenue and year* or *revenue and month*). To facilitate exploration we built several visualizations working in a coordinated manner, using the D3 visualization library [4]. In the next section we explain them in detail.

### Query Pattern Browser

The query pattern browser allows the user to select a template or a column semantics concept and explore the relative popularity of the selected concepts and templates in a coordinated manner. Initially, templates and concepts are ranked by their popularity across all patterns. For example, we see that the template *"How do the values of X compare by Y?"* is ranked at the top among all query templates. Likewise, *measure_quantity_amount* is the top ranked concept for column X. When the user selects either a template or a concept for either of the columns, the rest of unselected widgets update their ranking according to the selection. In Figure 1 we see that upon selecting a concept *measure_quantity_amount* in column X the rankings of the template and concepts in column Y are updated. While *"How do the values of X compare by Y?"* is still at the top, in column Y we see that *region* is the top semantic concept, meaning then when *measure_quantity_amount* is the first concept in any query the top query template is *compare_by* and top associated column Y concept is *region*. As a user selects combinations of query templates and semantic concepts per slot, the system displays sample queries from the query logs that fall into these semantics (not shown here due to privacy). This final step of exploration gives the user a clearer understanding of the quality of summarization given concept abstraction at various levels.

Throughout these widgets we use varying levels of opacity to convey abstraction level of a semantic concept, and font and element size to convey relative frequency of use in queries. Concepts in each column can be filtered by either keywords and range of abstraction level of the concept. This allows

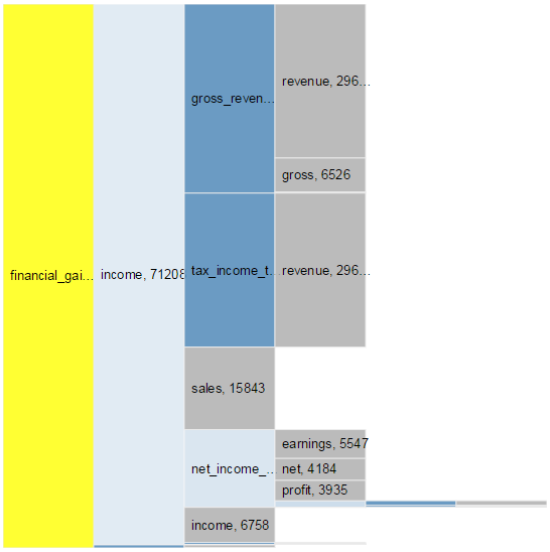


**Figure 2. Full breakdown of semantic type hierarchy by query count.**

the user easily focus on a particular semantic concept, say *financial_gain*, or just focus on concepts at high-levels of abstractions, for example by selecting a range from 5 to 8.

### Columns Semantics Hierarchy

When exploring semantic concepts in either column of the query it is important to understand what a particular concept means. One way to support that is to give examples of columns that match those semantics. Another way is to show the semantics hierarchy, in other words the composition of the concept.

When a concept is selected users can click to show the semantics hierarchy showing all hyponym concept branches in an icicle plot [6]. Elements to the left reflect concepts of higher abstraction and elements to their immediate right are hyponyms, i.e., descendant nodes within the WordNet hierarchy. Figure 2 shows such a hierarchical composition of a concept, i.e., *financial_gain*. As can be seen, *financial_gain* is composed of *income*, for the most part, and *income* is composed of *gross_revenue*, *tax_income*, *sales*, *net_income*, etc. The relative height of the concepts reflect the relative popular-

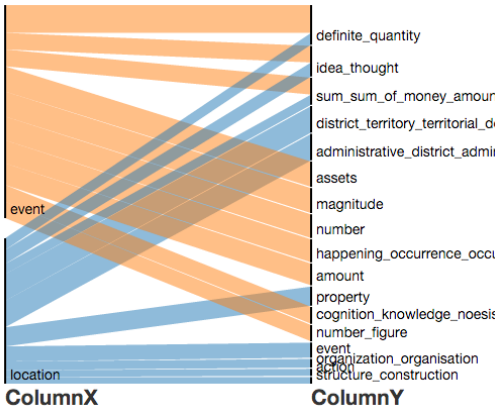


**Figure 3. Semantic type overlap and combination frequency by query count.**

ity for the current query selection criteria. This visualization gives users an intuitive and visual way of browsing of the semantic hierarchy under a particular concept and zooming in to a particular concept and its descendants on demand. We have chosen icicle plots for their effectiveness in displaying hierarchies from a bird's eye perspective, where size per element reflects frequency of use of a semantic type in a query.

### Column Semantics Concept Associations

While exploring the composition of column semantics is useful, exploring associations of such semantics is critical, especially for understanding how combinations of concepts are queried. To compare combinations of semantic concepts we use parallel sets. Parallel sets [15] is a visualization technique that adopts the layout of parallel coordinates but substitutes data points by frequency based representation. This feature is ideal to effectively convey relative frequency of combinations of selected semantic concepts, and combinations of all subconcepts therein. For example, having selected a template, the semantic concept *financial_gain*, and the semantic concept *time_unit_of_time*, the user can see frequency of combinations of subtypes of *financial_gain* and *time_unit_of_time* for the selected template. Figure 3 shows a parallel set view of two concepts (e.g., *event* and *location*) on a column X and the respective associations of concepts on a column Y. We see here that *event* is associated with *definite_quantity* more than *location* while *location* is associated with *property* concept. Exploring such associations allows the user to understand the relative rankings of combinations of concepts.

Additionally the tool offers inspection of the underlying hierarchies of semantic concepts ranked by frequency of use, and review of popularity of subtypes or subtrees is available on a point by point basis. A user can filter through available options by leveraging abstraction of semantic concepts, or select multiple semantic concepts to compare their interaction with other semantic concepts. For any abstraction level of semantics, the user can select as input multiple semantic concepts. All combinations involving selected semantic concepts (and their children), are displayed as parallel sets.

## EVALUATION

To understand the value of query semantics in regards to query recommendation we conducted an interview study with an IBM Watson Analytics technical lead, following an earlier meeting with several leads a couple of weeks prior. The purpose of the study was to understand the current process for query recommendation and obtain feedback on the utility of query semantics data and on the usability of the VIQS tool in regards to representation of and interaction with such data. After a brief 5-minute recap of the process for creating query semantics data we introduced the features of VIQS in a live demonstration. We spent all remaining time in a semi-structured question and answer format.

Our interview started with questions on the query semantics data. Clearly, the utility of query semantics data along with metadata on query popularity was seen as very useful for query recommendation, as he said: *"We absolutely want to use and to gather statistics learned from them, gather usage data and learn from it and apply it to recommendation."* In addition, the abstraction levels associated with each query semantics were also seen as an important feature, in particular to drive query understanding: *"I saw a breakdown that went from very specific all the way up to very abstract - I think that would be very useful. When people ask very vague questions we want to know what they are more likely to mean. I think we can use this data to definitely drive that."* However, he did raise concerns regarding potential bias in the data as the original query logs contained queries selected by the user from the set of queries recommended by the system to the user. While one can argue that a user's selection is a confirmation of user's approval of the recommended query to be a valid query on the data, a counter-argument is that the query is still a system recommended query - potentially not perfectly matching a user's desired question but as a good starting point which she can progressively tweak using the tool. As he said: *"We need to be able to differentiate when the user just picked something that we recommended blindly or whether they have gone and tweak[ed] things and modified things until they are happy and to learn strongly more from those."*

Overall, the tool is found be *"very useful"* and *"a good place to start"*, in particular, *"very valuable to understand the level of abstraction people are asking"*. The premise of the tool is seen as a utility to help build a limited ontology on the semantics of queries, as it was the practice: *"The expert system does have rules [about?] how it recommends.... Some of those rules are of the form, if we see location at any level in the concept hierarchy, if we see this, then we increase the score. [...] We created a very limited ontology for the things we would recognize and limited the noise that way."* The lead explained that their goal is to build more domain-specific ontologies, but explained that it is challenging as different words have different meanings: *"For example, there are terms in HR that are different from finance but we are not domain experts that is why we haven't dealt with that as much as we should have."* The lead expected VIQS to be particularly useful in that regards: "*The* (referring to VIQS) *tool would be very useful help to guide us through that manual process - at the moment we don't have the richness of the use case as well as we should.*" Referring to past attempts to create a rich ontology from WordNet he explained that it introduced a lot

of noise in the system because *"You are getting every possible meaning of the word but many of them don't even apply."*

### Insights for Query Recommendation

Below we report on our findings in regards to specific insights we identified from examining the query semantics data collected during a three-month period. When looking at the most popular query templates we see that *compare* queries take up about 37.8% of the total queries, followed by *breakdown* queries at 20.0%, *trend* at 13.9%, *values* at 11.2%, and *relationship* at 9.6%, while the remaining templates (*contribution*, *related*, and *number*) constituted 7.5% in total.

We examined the distribution of semantic concepts used across the board in all query templates. For both X and Y placeholders (referred to as column X and column Y for simplicity) *measure-quantity-amount* is the top semantic concept by far, accounting for 24.2% of column X and 18.1% of column Y in queries. However, beyond that ranking of concepts is different. While for X we see that *possession* (15.1%, particularly *assets* (14.8%)), *event* (13.4%), *property* (12.9%), *sum of money* (10.9%, particularly (*financial loss* (9.7%)), *location* (9.1%), and *activity* (8.6%) are top concepts, for Y we see concepts such as *social_group* (15.4%, particularly *organization* (12.2%)), *location* (14.8%, particularly *region* (12.3%)), *time* (11.3%), and *event* (11.3%).

We examined the semantic concepts for other query templates, such as *compare*, *breakdown*, *relationship*, and *trend*. We broke down the semantic concepts into two categories: high and low-level concepts, corresponding to abstraction levels 6 and above, and 5 and below, respectively. The purpose of this categorization was to limit the impact of high-level concepts on low-level concepts. While analysis of high-level concepts are expected to show gross patterns such as comparison between two quantitative measures, analysis of low-level concepts are expected to show more data-specific patterns such as trends of cost over time, perhaps even at finer levels such as over specific time periods (e.g., *year*, *month*, *quarter*).

When we examined high-level concepts in *compare* queries we seaw that not surprisingly *measure_quantity_amount* tops the list at 23.2% for column X. For column Y we see *social_group* at the top of the list at 17.4% (particularly *organization* contributing 14.7%) while *measure_quantity_amount* comes lower in the rank at 11.4% than *status-position* at 13.3%, *event* (13.3%), *category* (12.0%), *location* (11.7%). Other high-ranking concepts in X were *possession* (18.7%), *property* (12.7%), *financial_loss* (12.6%, *cost* contributing 12.4%), event (11.7%), *location* (9.3%, *region* contributing 6.5%), *activity* (9.1%), *feeling* (8.6%), *quality* (6.3%). These results confirm our expectation that concepts in column X tend to be more observable concepts (i.e., a quantity, or some measure of finance, activity, feeling or quality), while concepts in column Y generally represented control variables where people are interested in understanding observed concepts by some categorization (i.e., *group*, *category*, *location*, etc.).

When we examined top ranking low-level concepts in *compare* queries we saw concepts such as *sum_of_money* (9.5%, *payment* contributing 9.2%), *ratio* (9.2%), *time_period* (7.3%), *satisfaction* (7.0%), *income* (6.4%) in column X. For column Y we saw *enterprise* (8.2%), *management_direction* (8.0%), *reputation* (7.9%), *time_period* (7.2%), *music_genre* (6.8%), etc. We saw that in both columns there was no single concept significantly more popular than others (the percentages for top-k concepts while declining were fairly flat). Higher-ranking concepts also made sense, for example, several queries contained columns that mapped to money. The *ratio* concept probably ranked high because several columns included some sort of 'percentage of' some quantitative column. The *satisfaction* concept is also high probably due to survey datasets.

We did similar analysis on all query templates. *Breakdown* queries are found to be very similar to *compare* queries, where column X is mostly populated with concepts representing observable data and Y is more controlled data. *Relationship* queries on the other hand had evenly balanced concepts on both X and Y. In fact, the top-5 rankings for both columns are exactly the same with very similar percentages. Column X top concepts are *measure_quantity_amount* (23.8%), *possession* (15.9%), *assets* (15.4%), *event* (12.4%), *property* (12.2%), and *idea-thought* (8.5%). Column Y top concepts are likewise *measure_quantity_amount* (22.4%), *possession* (13.2%), *assets* (12.8%), *event* (12.5%), *property* (11.3%), and *sum_of_money* (8.7%). This makes sense as *relationship* queries seem commutative, where position within the query doesn't matter.

Analysis of *trend* queries revealed that while distribution of semantic concepts for column X was very similar to other query templates, column Y had some very high-ranking concepts with *measure_quantity_amount* at 45.2% and *time* 35.3% (*month* contributing 12.5%), while remaining referenced concepts such as *property* (12.0%), *location* (9.1%), *social_group* (8.5%), and *age* (7.3%) were subdued. *Values* queries behaved similarly, but in this case column Y top four concepts are *location* (55.8%) and related concepts such as *region*, *district* and *territory*, subsequent top four concepts are *social_group* (47.6%) and related concepts.

## DISCUSSION

In this paper we have introduced some of the unique challenges faced in query recommendation in a multi-user multi-database analytics platform like Watson Analytics. It is evident that state-of-the-art approaches to query recommendation are not applicable in such an environment. We use the popular lexical knowledge-base WordNet to tag datasets with concepts to aggregate and summarize the variety of column names referenced in queries across all the databases uploaded to Watson Analytics. In this way, we can generate multiple abstractions of queries that can be used to view and understand the types of queries asked in a principled way.

We introduce a first attempt at visual exploration of query semantics. With VIQS a user can gather insights in common patterns seen in the query logs spanning multiple datasets. Indeed, initial findings are intuitive, and confirm that query popularities over semantic summaries provide an important starting point for exploring new tables even before we have any query log that covers them. We discussed our tool with a technical lead, who confirmed the necessity for query recommendations in analytics platforms and the challenges faced in generating useful recommendations.

Currently the design and implementation of VIQS allows for exploration of query logs using abstract concepts all the way to the original queries performed over data in the Watson Analytics data lake. While WordNet makes sense as a first step towards concept tagging, in the future it could be useful to further narrow the scope of concept tagging of data tables themselves with particular domains, as was brought forth in our user interview. This could be done by leveraging upper merged ontologies (e.g., SUMO [19]) and their connected domain ontologies as well as using our tool to prune and aggregate such ontologies for specific domains as discussed in the interview. We believe involving the users more directly in query recommendation would prove very useful in increasing the quality of recommendations and building a better model.

In our view, the visual analytic approach to a fundamentally information management problem, query recommendation, is very promising. While size and complexity of the data may initially prohibit applying visual techniques, through appropriate aggregations and interactive methods for filtering supporting such model building activity is feasible. Complexity of the data, hierarchies of concepts, associations among concepts, popularity, levels of abstraction were an integral part of the problem. We believe the use of coordinated visualizations helped us achieve a reasonable interaction to support understanding of query semantics.

## CONCLUSIONS

In this work, we presented what is, to the best of our knowledge, a first case of visual analysis of structured data analysis query logs over a large repository of diverse datasets issued by multiple users. We illustrated some challenges that make exploiting these query logs hard including the heterogeneous nature of the data tables, the data domains, and the users. To address heterogeneity we tagged structured queries with semantic concepts, by leveraging a popular human-annotated lexical database, WordNet. We used a hierarchy of concepts to summarize the queries in the query logs and we provided a first attempt at visualizing and understanding queries in such logs, and discovering what the common querying patterns are. We applied well known visual analytics techniques to enable users effectively form hypotheses and generate insight from the complex task of understanding this rich and unique dataset.

## ACKNOWLEDGMENTS

Work was supported in part by NSERC.

## REFERENCES

1. J. Akbarnejad, G. Chatzopoulou, M. Eirinaki, S. Koshy, S. Mittal, D. On, N. Polyzotis, and J. S. V. Varman. Sql querie recommendations. *Proc. of VLDB Endowment '10*, 3(1-2):1597–1600, 2010.
2. J. Aligon, M. Golfarelli, P. Marcel, S. Rizzi, and E. Turricchia. Similarity measures for olap sessions. *KAIS*, 39(2):463–489, 2014.
3. D. A. Benson, M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers. Genbank. *Nucleic Acids Research*, 41(D1):D36–D42, 2013.
4. M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. 2011.
5. G. Chatzopoulou, M. Eirinaki, S. Koshy, S. Mittal, N. Polyzotis, and J. S. V. Varman. The querie system for personalized query recommendations. *IEEE Data Eng. Bull.*, 34(2):55–60, 2011.
6. F. Chevalier, D. Auber, and A. Telea. Structural analysis and visualization of c++ code evolution using syntax trees. In *IWPSE'07*, pages 90–97. ACM, 2007.
7. C. Collins, M. S. T. Carpendale, and G. Penn. Docuburst: Visualizing document content using language structure. volume 28, pages 1039–1046, 2009.
8. Data.gov. `http://www.data.gov/`.
9. M. Drosou and E. Pitoura. Redrive: result-driven database exploration through recommendations. In *Proc. of ACM CIKM'11*, pages 1547–1552, 2011.
10. C. Fellbaum. Wordnet: An electronic lexical database. 1998.
11. A. Giacometti, P. Marcel, and E. Negre. A framework for recommending OLAP queries. In *Proc. of DOLAP 2008*, pages 73–80, 2008.
12. M. Golfarelli, S. Rizzi, and P. Biondi. myolap: An approach to express and evaluate OLAP preferences. *IEEE TKDE'11*, 23(7):1050–1064, 2011.
13. H. Guo, S. R. Gomez, C. Ziemkiewicz, and D. H. Laidlaw. A case study using visualization interaction logs and insight metrics to understand how analysts arrive at insights. *Proc. of IEEE TVCG'16*, 22(1):51–60, 2016.
14. E. Kandogan, M. Roth, P. M. Schwarz, J. Hui, I. Terrizzano, C. Christodoulakis, and R. J. Miller. Labbook: Metadata-driven social collaborative data analysis. In *IEEE Big Data*, pages 431–440, 2015.
15. R. Kosara, F. Bendix, and H. Hauser. Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE TVCG'06*, 12(4):558–568, 2006.
16. V. A. Kulyukin, K. J. Hammond, and R. D. Burke. Answering questions for an organization online. In *Proc. of AAAI and IAAI '98*, pages 532–537, 1998.
17. V. Nastase. Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation. In *Proc. of EMNLP'08*, pages 763–772. Association for Computational Linguistics, 2008.
18. W. B. Paley. Textarc: Showing word frequency and distribution in text. 2002.
19. A. Pease, I. Niles, and J. Li. The suggested upper merged ontology: A large ontology for the semantic web and its applications. In *AAAI-2002 workshop on ontologies and the semantic web*, volume 28, 2002.
20. C. S. Ray, D. H. Goh, and S. Foo. *The Effect of Lexical Relationships on the Quality of Query Clusters*. 2006.
21. C. Sapia. PROMISE: predicting query behavior to enable predictive caching strategies for OLAP systems. In *Proc. of DaWaK'00*, pages 224–233, 2000.
22. S. Scott and S. Matwin. Text classification using wordnet hypernyms. In *Proc. ofCOLING/ACL Workshop on Usage of WordNet in NLP Systems*, pages 38–44, 1998.
23. A. Simitsis, G. Koutrika, and Y. E. Ioannidis. Précis: from unstructured keywords as queries to structured databases as answers. *VLDB J.*, 17(1):117–149, 2008.
24. F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: A large ontology from wikipedia and wordnet. *Journal of Web Semantics*, 6(3):203–217, 2008.
25. F. van Ham, M. Wattenberg, and F. B. Viégas. Mapping text with phrase nets. *IEEE TVCG'09*, 15(6):1169–1176, 2009.
26. Watson analytics. `http://www.ibm.com/analytics/watson-analytics/`.
27. M. Wattenberg and F. B. Viégas. The word tree, an interactive visual concordance. *IEEE TVCG'08*, 14(6):1221–1228, 2008.
28. J. Wen, J. Nie, and H. Zhang. Query clustering using user logs. *ACM TOIS'02*, 20(1):59–81, 2002.
29. D. Yang, E. A. Rundensteiner, and M. O. Ward. Nugget discovery in visual exploration environments by query consolidation. In *Proc. of ACM CIKM'07,*, pages 603–612, 2007.
30. X. Yang, C. M. Procopiuc, and D. Srivastava. Recommending join queries via query log analysis. In *Proc. of IEEE ICDE'09*, pages 964–975, 2009.
31. Q. Yao, A. An, and X. Huang. Finding and analyzing database user sessions. In *Proc. of DASFAA'05*, pages 851–862, 2005.