KOIOS: Utilizing Semantic Search for Easy-Access and Visualization of Structured Environmental Data *

Veli Bicer[1], Thanh Tran[2], Andreas Abecker[3], and Radoslav Nedkov[3]

[1] FZI Forschungszentrum Informatik, Haid-und-Neu-Str. 10-14, D-76131 Karlsruhe, Germany, bicer@fzi.de
[2] Institute AIFB, Geb. 11.40 KIT-Campus Sd, D-76128 Karlsruhe, Germany duc.tran@kit.edu
[3] disy Informationssysteme GmbH, Erbprinzenstr. 4-12, D-76133 Karlsruhe, Germany firstname.lastname@disy.net

Abstract. With the increasing interest in environmental issues, the amount of publicly available environmental data on the Web is continuously growing. Despite its importance, the uptake of environmental information by the ordinary Web users is still very limited due to intransparent access to complex and distributed databases. As a remedy to this problem, in this work, we propose the use of semantic search technologies recently developed as an intuitive way to easily access structured data and lower the barriers to obtain information satisfying user information needs. Our proposed system, namely KOIOS, enables a simple, keyword-based search on structured environmental data and built on top of a commercial Environmental Information System (EIS). A prototype system successfully shows that applying semantic search techniques this way provides intuitive means for search and access to complex environmental information.

1 Introduction

As environmental issues become a hot topic for the general public, we perceive that the amount of publicly available environmental data is also continuously growing on the Web. Over the last ten years, public access to environmental data is highly encouraged by the governments since they have recognized that environmental information could have a profound impact on our ability to protect the environment [7]. Starting from the Directive 2003/4/EC, for instance, the European Union grants public access to environmental data. *PortalU*[4] in Germany, *Envirofacts*[5] in the USA or *EDP*[6] in the UK are just few examples that provide access to large volumes of environmental data as a result of recent activities. Besides, environmental data are also made accessible as a part of the Linking Open Data (LOD)

* This research was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference 01MQ07012. The authors take the responsibility for the contents.

[4] http://www.portalu.de/
[5] http://www.epa.gov/enviro/index.html
[6] http://www.edp.nerc.ac.uk

project in a structured format (RDF) with the idea of linking environmental data in an international context of cooperating governmental authorities [15]. Thus, previously local databases of environmental data have become part of the LOD cloud of datasets, enabling the active dissemination of environmental information to the masses.

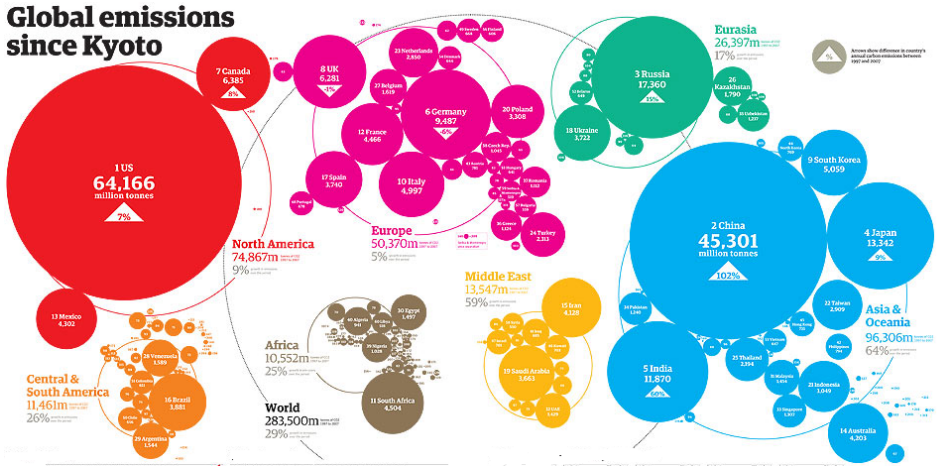


Fig. 1: An illustration of carbon emissions around the World (source: Copenhagen Climate Council).

Despite the increasing importance and availability, the commercial and societal impact of open environmental data is still very limited to the end-users. This has a number of reasons, but one of them is certainly the largely intransparent access to complex databases. This obviously holds true for interested citizen and companies, but even for employees of public authorities in a different domain, the heterogeneity and distribution of environmental data is often overwhelming. Hence, user-friendly and powerful search interfaces are a must-have in this area. For example, assume for a moment that you are searching carbon (CO) emission values that are highly critical figures considering their impact on the climate change and global warming. Each year several organizations review the underlying consumption data for petroleum, natural gas and coal worldwide and estimate our CO emission values. By using a classical Web search and keyword queries, it is quite easy to obtain the country-specific figures as shown in Figure 1 as these are just parts of the documents in which the keywords occur. However, a more detailed search on the CO emission values such as "CO emission values around Karlsruhe area in Germany" or obtaining more analytical results based on a particular year, or emission type (e.g. by industry, by transportation) requires more advanced search on structured data. This search paradigm however, assumes users to be an expert of the underlying data and domain, or design of proprietary interfaces to access data. The main challenge here is to provide ordinary users an easy-to-formulate queries (i.e. keyword query) and provide complex structured results in return to satisfy their

information needs. According to the survey in [6], the percentage of people who used the Internet to find environmental information (49%) is significantly lower than those with frequent access to it (69%).

Recently, semantic search approaches to enable keyword search on structured data has gained a lot of interests as keywords have proven to be easy for the user to specify, and intuitive for accessing information. By utilizing lightweight semantics of the underlying RDF data, keyword search can help to circumvent the complexity of structured query languages, and hide the underlying data representation. Without knowledge of the query syntax and data schema, even the non-technical end-users can obtain complex structured results, including complex answers generated from RDF resources [17, 19]. To this end, semantic search in that sense can provide the means to simplify the search on environmental data, allowing the users to access rich information with less effort, and lower the access barriers resulting from the complex interfaces of current systems.

In this work, we present a novel semantic search system, namely KOIOS, that provides semantic search capabilities on structured data with the aim of easy-access to rich environmental information. Our contributions mainly include: 1) enabling keyword search as an intuitive mechanism to search environmental data, 2) interpretation of user's possible information needs via the query translation from keyword query to structured query by utilizing the underlying schema structure and data, 3) offering dynamic and flexible faceted search to allow the user to specify his/her further preferences, and 4) integration to a commercial, Web-based Environmental Information System (EIS) for user-friendly presentation of information using visualization components such as maps, charts, tables, etc.

Structure. In Section 2, we introduce an overview of KOIOS system. Section 3 presents the KOIOS semantic search process detailing the steps starting from the specification of the keyword query to the final display of the search results. In Section 4, a prototype implementation of the system is presented. After a discussion of related work in Section 5, we conclude in Section 6.

2 KOIOS Overview

The KOIOS system provides semantic search capabilities over structured data that is either present in relational databases or RDF repositories. Figure 2 depicts an overview of the KOIOS system. KOIOS operations can be divided into two main stages: *preprocessing* and *search*. Preprocessing is an offline stage not visible to the users that mainly creates three special search indexes out of the structured data of a given database or RDF store. The core index directly created from the data is called *data index*, which is a graph-based representation of the data implemented as an inverted index, and optimized for efficient access. Based on the data index, two other indexes are also created: the *keyword index* is mainly designed for IR-style access that captures the unstructured part of the data, and the *schema index* is extracted from the data, representing classes and relationships among them.

The second stage, search, is the actual part in which the system interacts with the user. User specifies his/her information with a short keyword query and considered as a set of keywords, $Q = \{q_1, q_2, ..., q_n\}$. Based on this query, the system first discovers possible keyword elements using the keyword index to find

particular tuples (entities) in the data in which one of the keywords occur. A number of keyword tuple sets are created for each keyword in the query. These sets are then combined with the schema information resulting in an augmented schema graph, which represents the query space. By exploring this graph, a number of structured query graphs are constructed, each of which can be executed on the data index to find relevant results. This part of the search stage mainly interprets the user's possible information needs in terms of structured query graphs, and computes their corresponding result sets.

Based on the outputs of this stage, KOIOS generates a number of facets to facilitate further interactions and refinements of query and results. It uses a faceted search interface to present the possible categories (facets) and values generated from the underlying results. This helps the user to refine his/her query. Further, additional user preferences can be incorporated in order to obtain more precise results. In particular, KOIOS maps user choices and preferences to an internal representation, called *selectors*, which are coarse-granular query templates that are used to generate more precise queries to be executed over the database. This selector mechanism is also employed to select appropriate presentation components. The results are presented to the user via an integrated EIS interface.

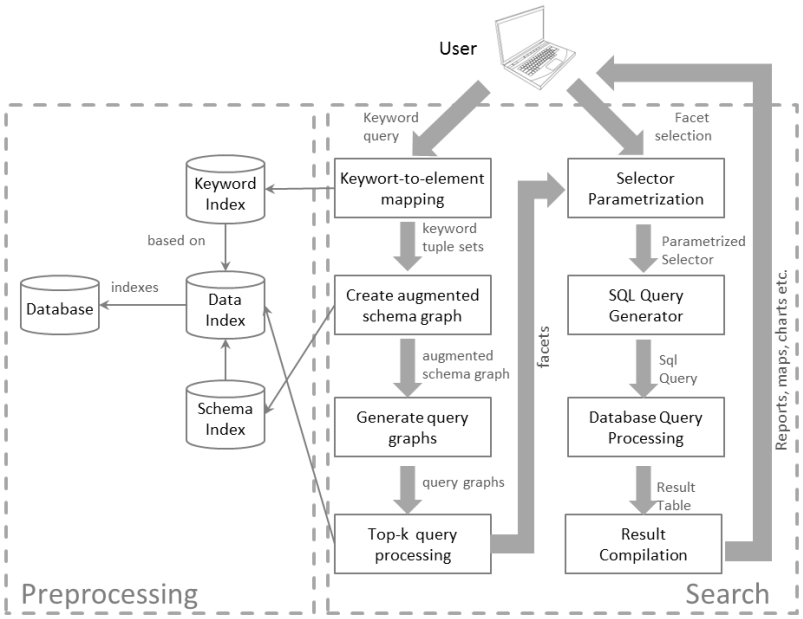


Fig. 2: Overview of KOIOS System

In overall, KOIOS minimizes the inherent complexity of searching structured data by guiding the user through the search process via analyzing the underlying

data and schema structure. This significantly minimizes the effort and cognitive complexity in the search process.

3 KOIOS Semantic Search Process

In this section, we present technical details of the semantic search process over environmental data. For the sake of presentation, we decompose the overall process into three steps: 1) Indexing, 2) keyword query interpretation and structured query generation, and 3) faceted search and selectors.

3.1 Indexing

In KOIOS, we apply a preprocessing step on the data to create index structures that help to perform the search functionalities more efficiently. Generally speaking, the underlying data can be conceived as a directed labeled data graph $G = (V, E)$, where V is a disjoint union ($V = V_R \uplus V_A$) of resource nodes (V_R), and attribute nodes (V_A), and $E = E_F \uplus E_A$ represents a disjoint union of relation edges also called foreign key edges (E_F) that connect resource nodes, and attribute edges (E_A) that link between a resource node and an attribute node. This model closely resembles the graph-structured RDF data model (omitting special features such as RDF blank nodes). The intuitive mapping of this model to relational data is as follows: a database tuple captures a resource, its attributes, and references to related resources in the form of foreign keys; the column names correspond to edge labels.

In order to perform the search steps in an efficient way, we preprocess the data graph to obtain the data index, which is actually a number of inverted indexes that store data in the form of $subject \rightarrow predicate \rightarrow object$ triples. In particular, each inverted index returns a specific element of the triple given the other pair of elements. Based on this index, the keyword index that is used for keyword-to-element mapping is created. Conceptually, the keyword index is a keyword-resource map and used for the evaluation of a multi-valued function $f : V \rightarrow 2^{V_R}$, which for each keyword $k_i \in V$ in the vocabulary, returns the set of corresponding graph resources V_R (i.e. keyword elements). In addition, a lexical analysis (stemming, removal of stopwords) as supported by standard IR engines is performed on the attributes of resources in order to obtain terms. Processing attributes consisting of more than one word might result in many terms. Then, a list of references to the corresponding graph elements is created for every term.

For exploration, a schema index is constructed, which is basically a summary of the original data graph containing structural (schema) elements only. In essence, we attempt to obtain such a schema from the data graph instead of assuming a pre-given schema. The computation of the schema index follows straightforwardly from and is accomplished by a set of aggregation rules presented in previous work [18], which compute the equivalence classes of all resources belonging to one class and project all edges to corresponding edges at the schema level with the result that for every path in the data graph, there is at least one path in the schema index (while this is not the other way round).

At the time of query processing, this schema index is augmented with keyword elements obtained from the keyword-to-element mapping. Since we are interested in the top-k results, the index elements are also augmented with scores. While scores associated with structure elements can be computed off-line, scores of keyword elements are specific to the query and thus can only be processed at query computation time.

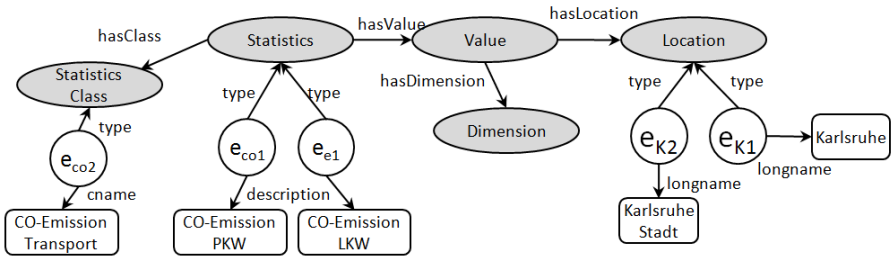


Fig. 3: Augmented schema graph showing schema elements (in gray), associated keyword elements (circles) and their corresponding attributes (rectangles).

3.2 Keyword Query Interpretation and Structured Query Generation

The goal of this step is to interpret the keywords entered by the user using the index structures created before, and to generate a number of structured queries. These queries are basically conjunctions of triple patterns, forming graph patterns corresponding to the Basic Graph Pattern feature of SPARQL. Typically, due to the ambiguity inherent in keyword queries, such an interpretation is not unique. Therefore, we rely on a top-k procedure to generate candidate interpretations and obtain the possible results that best match the user information need. We build on our previous work [17] on translating keyword queries into structured queries based on a graph-exploration technique. For this purpose, we consider available knowledge bases and data as data graphs as defined previously. (Intuitively, this graph

The computation of structured queries as interpretations of the user keywords involves three tasks: 1) detecting the keyword elements as resources in the graph containing at least one of the query keywords, 2) creation of an augmented schema graph, 3) graph exploration and structured query generation and 4) top-k query processing to compute the best queries. Specific concepts and algorithms for these tasks have been introduced in [17].

Keyword-to-element mapping: We rely on the keyword index to map keywords to elements of the data graph, which in our approach might be classes, foreign key relationships, attributes, and attribute values. IR concepts are adopted to support an imprecise matching that incorporates syntactic and semantic similarities. As a result, the user does not need to know the exact labels of the data elements while doing keyword search. Each element finally returned is associated with a

score measuring the degree of matching, which is later used for ranking possible interpretations. For each query keyword $q_i \in Q$, we define a set of candidate keyword elements $V_i = \{v_1, .., v_n\}$ that are potential elements in the data graph that the user is looking for.

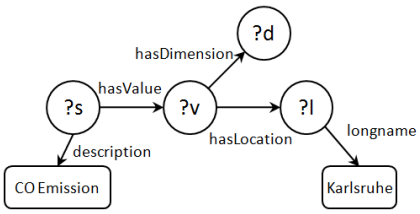
Augmented schema graph: Given the sets of keyword elements for the query keywords, the query search space contains all the elements that are necessary for the computation of possible interpretations. This mainly includes all keyword elements and the corresponding classes, foreign key edges, and attribute edges of the data graph. It has been shown that keyword search is most efficient when the exploration for possible interpretations is performed on an augmented schema graph, instead of using the entire data graph (c.f. [17]). The schema graph can be trivially obtained from the class and property definitions in the data, or might be pre-given as an ontology. From experiences with Web data, we know that pre-given schemas are typically incomplete and often do not reflect the underlying data as some schema elements may actually not be instantiated in the data. Therefore, we additionally apply techniques for computing schema graphs automatically. In particular, a schema graph is derived from the data using the aggregation rules as described in [17] (during preprocessing).

Figure 3 illustrates the query space constructed for our example keyword query "karlsruhe co emission". It consists of (the fragment of) a schema graph (nodes in gray), keyword elements found in the previous step for different keywords, and the corresponding attributes of those elements. Note that the augmented schema graph consists of all keyword elements and all the possible paths between them that can be found in the schema.

Graph exploration and query translation: Given the augmented schema graph, the remaining task is to search for the minimal query graphs in this space. Informally, a query graph is a matching subgraph of the augmented schema graph, such that for every keyword of the user query, it contains at least one representative keyword-matching element, and (2) the graph is connected, i.e. there exists a path from every graph element to every other graph element. A matching query graph is minimal if there exists no other query graph with a lower score. This procedure starts from the keyword elements and iteratively explores the query space for all distinct paths beginning from these elements. During this procedure, the path with the highest score so far is selected for further exploration. At some point, an element might be discovered to be a connecting element, i.e. there is a path from that element to at least one keyword element, for every keyword in the user query. These paths are merged to form a query graph. The explored graphs are added to the candidate list. The process continues until the upper bound score for the query graphs yet to be explored is lower than the score of the k-ranked query graph in the candidate list. An example query graph that can be found through exploration along these paths is shown in Figure 4.

Top-k Query Processing: Query translation results in a set of query graphs (not only one), each of which can be a potential representation of the user's information need. In fact, at this point we are not interested in all the final results to be retrieved from the database using the computed queries, but only in the top-k results, based on which facets will be constructed in the next step. Thus, ranking the top results (e.g. answers to SPARQL query) w.r.t. their relevance to the initial

keyword query Q is more important. The goal is to summarize these results as facets and to use the resulting facets for enabling additional refinement. For this, we query our data index with a retrieval algorithm as sketched in our previous work [17] to get top results for each query graph. In the next step, we discuss how facet values are generated from this initial query run on the data.




```
Select ?s,?v,?d,?l
WHERE {
  ?s ns:description "CO Emission" .
  ?s ns:hasValue ?v .
  ?v ns:hasDimension ?d .
  ?v ns:hasLocation ?l .
  ?l ns:longname "Karlsruhe"
}
```

(b) SPARQL Query

Fig. 4: a) An example query graph and b) the corresponding SPARQL query.

3.3 Faceted Search and Selectors

The query graphs and their corresponding result sets are possible interpretations of the query. However, instead of assuming that a query graph is a direct representation of the user's information need, we utilize it in KOIOS to bridge the gap between the keyword query and the actual information need. For this purpose, we support a second round of user interaction as shown in Figure 2. Basically, we aim to increase the precision of the final search results, while providing an intuitive and easy-to-use way for the user to specify further preferences. We propose to use 1) facets generated from a number of query graphs and their result sets, as well as 2) selectors that are parameterized, pre-defined query templates, run against the database in the back-end, and map to final visualization elements to be displayed to the user.

Facets: In IR, faceted search mostly refers to techniques for accessing a collection of information represented using facets, allowing users to explore by filtering available information [9]. In this setting, facets correspond to the attributes (called facet categories) as well as possible attribute values (facet values) that are common to a set of resources (top-k results computed previously). Traditionally, they are mostly derived by an analysis of the text or from pre-defined fields in a database table. A shortcoming of this sort of faceted search is its basic data model, where facets are associated with sets of values from independent facet hierarchies. This model is too restrictive for some real-world data. A more appropriate faceted search solution for our environmental data scenario should provide richer insights into the data and the ability to perform flexible and dynamic aggregation over faceted data [1].

In our approach, we use the generated query graphs to construct dynamic facets for a particular search session to realize such a functionality. Given a query

graph, we consider every variable binding (e.g. $?s$, $?v$, $?l$, $?d$ in Figure 4-a) as a possible candidate for a facet category. We utilize the corresponding types of these variables to retrieve their particular descriptive attributes. For example, the *Statistics* class of the variable $?s$ is used to retrieve the descriptive attribute "description". Then a facet category is created with the name *Statistics.description* for this attribute. As we obtain the top-k entities of this query graph in the last step of the previous section, the facet values are generated by clustering values that are specific to the *description* attribute of those entities. For this particular attribute, the possible facet values are shown by our prototype system in Figure 6. Note that an attribute does not need to be a part of the query graph in order to be utilized as a facet category. Also, for example, for the *Value.Year* attribute of the *?v* variable, we further generate facet values representing possible years (e.g. 2000, 2005 etc.), which help the user to refine the search based on a restricted set of relevant options.

Selectors: Using the facet categories and values chosen by the user, the system can identify a number of stored, semantically indexed *selectors*, which are parameterized, pre-defined query templates finally used for accessing the data sources in the back-end. In our architecture, a selector contains a variety of information and have the core functionality to map the results to visualization elements such as tables, charts, or maps. Basically, a selector has three types of parameters:

1. *Data parameters.* These parameters represent particular attributes to specify the scope of the selector for a particular information need. Some of the parameters (if not all) can be initialized to a particular value via the facets. That is, facets categories and values are mapped to data parameters of the selector that correspond to the same attributes. For the other data parameters that are not initialized this way, a SQL query is constructed by the system to retrieve the corresponding data directly from the database.
2. *Query parameters.* These are parameters pre-defined for each selector, indicating GROUP-BY and SORT statements, which are finally used for generating the SQL query.
3. *Visualization parameters.* In our system, the query results are visualized using different means, each one of them is derived from the visualization parameters of the selector. In particular, these parameters capture the visualization or presentation type (data value, data series, data table, map-based visualization, specific diagram type, etc.) for selectors' results.

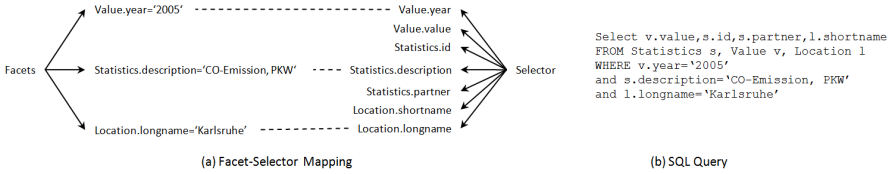


Fig. 5: (a) Mapping the facet values to the corresponding selector parameters, (b) the generated SQL query.

For example, Figure 5-a illustrates some possible facet values for our running example and their mappings to one of the selectors. As indicated, a selector may include more parameters than the facets. Thus, a selector can be flexibly initialized in different ways. Based on different facet categories and values, the system can generate a variety of initialized selectors, each corresponds to a different information need of the user. This way, with a relatively small number of selectors, the system can respond to a large number of queries. In addition, we also check the conformity of a possible facet selection of the user to a number of selectors available in the system to dynamically eliminate the selectors that can not answer a particular query. A selector is considered as non-conforming if it does not include any one of the facet categories (attributes) in its data parameters, and therefore is removed. In our prototype system, a number of selectors and their corresponding visualization components are automatically displayed to the user based on his/her selection of facets as shown in Figure 6.

Based on a particular initialization of selector, the system generates a SQL query, accesses the database and retrieves the results. An example SQL query is shown in Figure 5-b. Note that all the initialized values are captured in the WHERE clause whereas the non-initialized parameters are included in the SELECT statement of the query. The retrieved data are then displayed to the user as a final result based on his/her selection of visualization type (e.g. tables, maps, charts etc.).

4 Implementation

A prototypical KOIOS system has been implemented in Java within the scope of the German Internet project *THESEUS* [7]. For indexing the data, we use the open-source IR engine Lucene Framework [8] from Apache Software Foundation [9]. In addition, the system is integrated with a commercial EIS application Cadenza[10] for the management of selectors and for visualizing the results. In the current prototype, we are using the environmental data of the German state Baden-Württemberg, collected during the years from 1990 to 2006. The demo of the system is reachable at http://krake05.fzi.de:8888/koios.

In Figure 6, we can see the start page (in German) of the KOIOS semantic search engine. The user can type a keyword query using a text field similar to the one provided by classical Web search engines that most users are familiar with. In the initial execution of the query, the system performs the aforementioned steps of query translation and displays users the facets on the left hand side of the page together with the possible visualization options on the right. Common types of visualization options are grouped together as tables, charts, or maps. Besides, the system also displays the base selectors suitable to the query and selected facets in order to give the user the flexibility to specify his/her information need in a more fine-grained way using the functionalities of the underlying EIS Cadenza system.

[7] http://theseus-programm.de/
[8] http://lucene.apache.org/
[9] http://www.apache.org/
[10] http://www.disy.net/produkte/cadenza.html

Fig. 6: KOIOS search interface and facets generated for the query "karlsruhe co emission".

After the selection of facets and visualization type, the system displays the final results to the user using the Web version of the Cadenza system. Figure 7 shows the search result displayed as a chart for our running example query "karlsruhe co emission". As shown in the figure, the parameters selected include a number of possible emission types (e.g. industry, transportation PKW, LKW etc.), the year 2005, and Karlsruhe and Karlsruhe Stadt (i.e. Karlsruhe city center) for location. The system offers a number of visualization capabilities plus the option to store the chart into a file for later use. In addition, the resulting data can easily be visualized by another type of presentation module without the need for re-issuing the query thanks to the selector mechanism that can generate a variety of options from a single source query.

Another visualization option that displays the aggregated data as a map is shown in Figure 8. As in the case of chart display, the parameterization of this type of visualization is also done via facets in a dynamic fashion. Starting from a keyword query "karlsruhe co emission", the user can obtain a variety of results when employing KOIOS's semantic search capabilities. In this type of visualization, the configuration options such as coloring scheme on the right hand side of the map also provides additional capabilities to the user to perform further refinements.

5 Related Work

Our research relates to work from three major areas, namely (1) environmental data, (2) semantic search on structured data, and (3) faceted search.

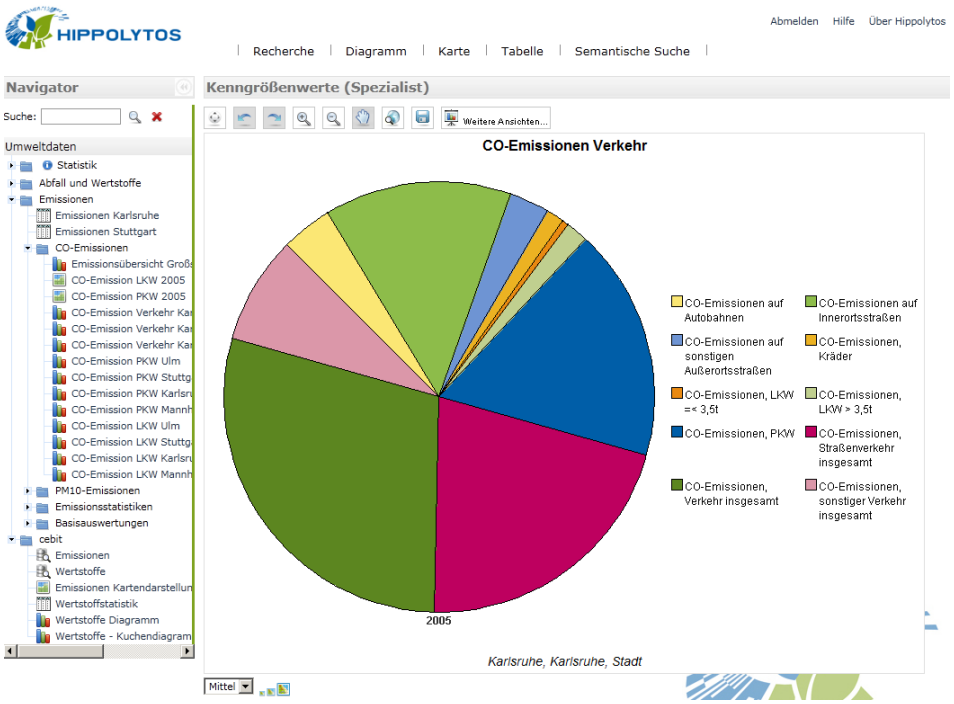


Fig. 7: Chart showing the CO emission values around Karlsruhe in 2005.

Environmental data on the Web: As environmental issues secured their position on a global level, environmental information has started to be considered as a public asset. As a consequence, governments and other administrative units have become more active in promoting access to environmental data as a mean to improve public participation in environmental decision making and awareness of environmental issues [7]. According to the survey conducted in [6], the major environmental information needs relate mainly to the people's everyday activities, which can be seen as a mix between livelihood issues, quality of life, and health issues. This mainly includes easy-access to information about public transport, air quality, water quality, traffic, noise, and toxicity. Traditionally, environmental data are managed with the help of Environmental Information Systems and Environmental Decision Support Systems (EDSS) [3, 4]. Recently, there is an abundance of environmental information on the Internet, from raw data which are broadcasted directly from monitoring stations, to politically charged information made available by specific interest groups [11, 20, 16, 14, 13]. Although the use of EIS and EDSS has a long history, easy-access to environmental data by the mass users has become a more prevalent problem, as public interest in these data increases. Previous works mostly provide proprietary interfaces to search and visualize data that are geared towards expert usage, and less tailored to the needs of the non-technical end users.

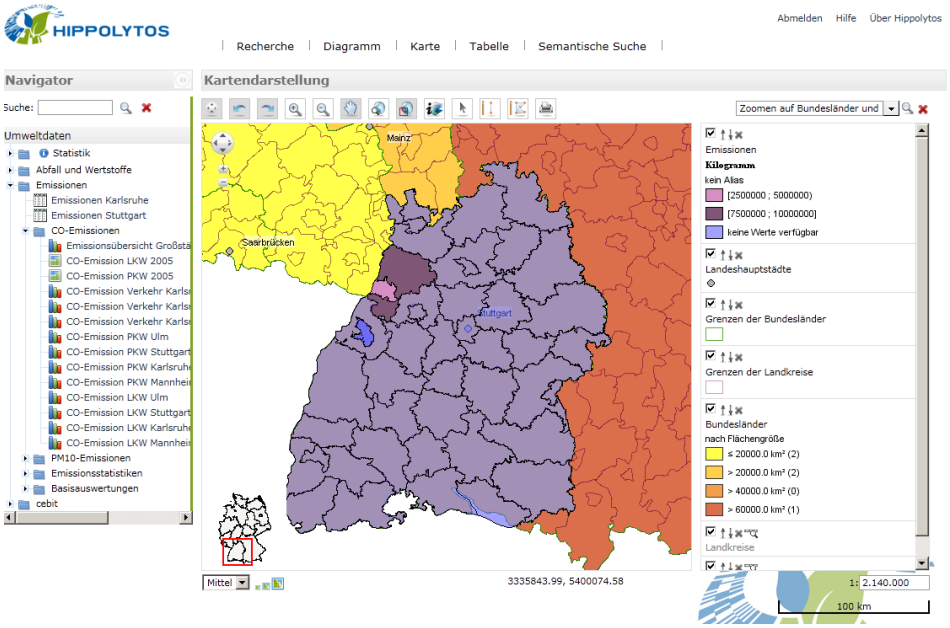


Fig. 8: Map showing the CO emission values around Karlsruhe in 2005.

Semantic search on structured data: Finding and ranking relevant resources is the core problem in the Information Retrieval community, for which different approaches have been investigated. Clearly, the main difference of keyword search on structured data and the traditional keyword search on documents is that instead of one single document, structured data may encompass several resources (e.g. database tuples, documents, RDF resource descriptions) that are connected over a possibly very long path of relationships. [19] provides a review of different semantic search tools and focuses on different modes of user interaction. Compared with other modes of interaction (form-based, view-based, or natural language), the advantages of keyword-based querying lie in its simplicity and the familiarity most users already have with it. The problem of keyword queries on structured data has been studied from two different directions: 1) computing answers directly through exploration of substructures on the data graph [10, 8, 12] and 2) computing queries through exploration of a query space [17]. It has been shown in [17] that keyword translation operates on a much smaller query space, and is thus efficient. Besides, the structured queries presented to the user help in understanding the underlying data (answer) and allow for more precise query refinement. We follow the second line of work to keyword search and adapt it to the problem of searching environmental data. In addition, in order to rank the final retrieved results from the databases, we recently proposed a relevance model based ranking support [2].

Faceted Search: Faceted search is increasingly used in search applications, and many Websites already feature some sort of faceted search to improve the precision of search results. A crucial aspect of faceted search is the design of a user interface, which offers these capabilities in an intuitive way. This has been studied

by [9, 21, 1] and applied in systems like Flamenco[11], Exhibit[12] or Parallax[13]. In a Semantic MediaWiki context, this paradigm has been applied by "Ask the Wiki" for browsing Wiki pages along semantic facets. [5]. Another cornerstone of faceted search is the question what is actually used as facets and if they are hierarchical or multidimensional, which obviously depends on the data corpus and its structure. Flamenco and Exhibit require a predefined set of properties for every data item, and then allows browsing along the values of these properties. We actually use the schema and query graphs for facet construction, and thus dynamically determine which facets should be present in a generic fashion. The diversity of query graphs results in a multidimensional facet generation, that user can refine by considering different aspects (e.g. time, location, category etc.). We also precompute the possible values of facets, which serve as feedback to the user and offer guidance for the underlying data and possible selectors to be chosen.

6 Conclusion and Future Work

Summary. We have sketched the functionalities and discussed the realization of KOIOS, a semantic search engine over structured environmental data. The goal is an "intuitive and simple way for information access" for emerging environmental data on the Web. KOIOS provides a Google-like, simple keyword-based query interface, which automatically finds and instantiates available selectors and thus automatically configures appropriate structured queries to be processed against the back-end data sources. It does not require pre-specified knowledge in the form of a schema or ontology, but instead, automatically computes schema graphs from the underlying data in order to find possible query interpretations. The results are provided to the user via a faceted-search interface, which facilitates further refinement in order to obtain more precise search results.

In fact, our approach differs from the use of a full semantic infrastructure that comes with expressive ontologies, SPARQL query processing and reasoning capabilities. In this regard, our work mainly incorporates an IR-based approach to infer the users' information needs, and to retrieve, rank and visualize the relevant data for those needs. We consider that in comparison, our approach offers a number of benefits: First, it removes the need to specify an expressive ontology capturing all the domain knowledge which is not easily attainable for various scenarios in the environmental domain. In addition, most of the Semantic Web approaches still rely on the definition of structured queries (e.g. SPARQL) that is not so easy-to-formulate for ordinary users as we discussed above. Finally, ranking support is an important element of IR-based approaches that we incorporate as a further extension to our approach [2].

Status and Future Work. The prototype system is built on top of a commercial EIS called Cadenza and demonstrated over real-world environmental data of the German state Baden-Württemberg. It shows for realistic data volumes and schema sizes that it is possible to deliver reasonable results with acceptable performance. The evaluation of the result quality is yet to be conducted through further

[11] http://flamenco.berkeley.edu/
[12] http://www.simile-widgets.org/exhibit/
[13] http://www.freebase.com/labs/parallax/

experiments. The ranking heuristics are most crucial, and particularly requires attention and detailed investigation in future work. Further, while the prototype can be seen as a study of technical feasibility, this work yet lacks evaluation from a usability point of view.

References

1. O. Ben-Yitzhak, N. Golbandi, N. Har'El, R. Lempel, A. Neumann, S. Ofek-Koifman, D. Sheinwald, E. Shekita, B. Sznajder, and S. Yogev. Beyond basic faceted search. In *Proceedings of the international conference on Web search and web data mining*, pages 33–44. ACM, 2008.
2. V. Bicer, T. Tran, and R. Nedkov. Ranking support for keyword search on structured data using relevance models. In *Proceedings of the 20th ACM conference on Information and Knowledge Management (CIKM 2011)*. ACM, 2011.
3. R. Denzer. Generic integration of environmental decision support systems-state-of-the-art. *Environmental Modelling & Software*, 20(10):1217–1223, 2005.
4. O. El-Gayar and B. Fritz. Environmental management information systems (emis) for sustainable development: a conceptual overview. *Communications of the Association for Information Systems*, 17(1):34, 2006.
5. P. Haase, D. Herzig, M. Musen, and T. Tran. Semantic wiki search. *The Semantic Web: Research and Applications*, pages 445–460, 2009.
6. M. Haklay. Public environmental information: understanding requirements and patterns of likely public use. *Area*, 34(1):17–28, 2002.
7. M. Haklay. Public access to environmental information: past, present and future. *Computers, Environment and Urban Systems*, 27(2):163–180, 2003.
8. H. He, H. Wang, J. Yang, and P. S. Yu. Blinks: ranked keyword searches on graphs. In *SIGMOD Conference*, pages 305–316, 2007.
9. M. Hearst. Design recommendations for hierarchical faceted search interfaces. In *ACM SIGIR Workshop on Faceted Search*, pages 1–5. Citeseer, 2006.
10. V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar. Bidirectional expansion for keyword search on graph databases. In *VLDB*, pages 505–516, 2005.
11. F. Kruse, S. Uhrich, M. Klenke, H. Lehmann, C. Giffei, and S. T "opker. Portalu®, a tool to support the implementation of the shared environmental information system (seis) in germany. In *European conference of the Czech Presidency of the Council of the EU TOWARDS eENVIRONMENT-Opportunities of SEIS and SISE: Integrating Environmental Knowledge in Europe, Prague*, 2009.
12. G. Li, B. C. Ooi, J. Feng, J. Wang, and L. Zhou. Ease: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data. In *SIGMOD Conference*, pages 903–914, 2008.
13. R. Mayer-Foll, A. Keitel, and W. Geiger. Uis baden-wuterttemberg. projekt aja. anwendung java-basierter und anderer leistungsfahiger losungen in den bereichen umwelt, verkehr und verwaltung. phase v 2004. *Wissenschaftliche Berichte, FZKA-7077*, 2004.
14. W. Pillmann, W. Geiger, and K. Voigt. Survey of environmental informatics in europe. *Environmental Modelling & Software*, 21(11):1519–1527, 2006.
15. M. Ruther, T. Bandholtz, and A. Logean. Linked environment data for the life sciences. *Arxiv preprint arXiv:1012.1620*, 2010.
16. F. Shrode. Environmental resources on the world wide web. *Electronic Green Journal*, 1(8), 1998.

17. T. Tran, H. Wang, S. Rudolph, and P. Cimiano. Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, pages 405–416. IEEE, 2009.
18. T. Tran, L. Zhang, and R. Studer. Summary models for routing keywords to linked data sources. *The Semantic Web–ISWC 2010*, pages 781–797, 2010.
19. V. Uren, Y. Lei, V. Lopez, H. Liu, E. Motta, and M. Giordanino. The usability of semantic search tools: a review. *The Knowledge Engineering Review*, 22(04):361–377, 2007.
20. T. Vogele, M. Klenke, F. Kruse, H. Lehmann, and T. Riegel. Easy access to environmental information with portalu. *Proceedings of EnviroInfo2006, Graz, Austria*, 2006.
21. K. 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.