Query Results Clustering by Extending SPARQL with CLUSTER BY

Agnieszka Ławrynowicz

Institute of Computing Science, Poznan University of Technology,
ul. Piotrowo 2, 60-965 Poznan, Poland
alawrynowicz@cs.put.poznan.pl

Abstract. The task of dynamic clustering of the search results proved to be useful in the Web context, where the user often does not know the granularity of the search results in advance. The goal of this paper is to provide a declarative way for invoking dynamic clustering of the results of queries submitted over Semantic Web data. To achieve this goal the paper proposes an approach that extends SPARQL by clustering abilities. The approach introduces a new statement, CLUSTER BY, into the SPARQL grammar and proposes semantics for such extension.

1 Introduction

Dynamic clustering is a way of organizing search results that has gained an attention from academia [12,13] and from commerce. The technique of dynamic clustering consists on grouping search results into clusters generated from the search results themselves. The user after submitting a query to the search engine is often faced by a large number of search hits, and forced to further perform query results exploration to find out what the data source contains. Therefore, the clusters may help him/her more quickly and easily understand the retrieved results, and save the time otherwise spent on analysing them.

The task of dynamic clustering has been commonly addressed in the context of clustering textual Web search results. For instance, Vivisimo/Clusty[1] is an example of a successful commercial application of this idea. This paper, however, concentrates on the novel task of clustering the results of queries submitted to the structured Semantic Web [1] data. The proposition of applying the functionality of Web search clustering engines in the context of the open and distributed Semantic Web environment is motivated by the fact that although the Semantic Web data is structured, the user may not know the structure and granularity of the data apriori. Therefore *data retrieval* from the Semantic Web may be more of the spirit of *information retrieval* from unstructured text, rather than the retrieval from local, relational databases.

The goal of this paper is to enable declarative means for invoking query results clustering. These declarative means are supposed to provide a framework to which clustering methods and algorithms may be plugged in. This paper contributes to achieving this goal by proposing to extend the standard Semantic

[1] http://clusty.com

R. Meersman, P. Herrero, and T. Dillon (Eds.): OTM 2009 Workshops, LNCS 5872, pp. 826–835, 2009.

Web query language *SPARQL* [16] with CLUSTER BY statement that enhances SPARQL with grouping capabilities. The extension adds new syntax elements to the SPARQL grammar that facilitate clustering of query results. The paper proposes the grammar, as well as the semantics for this extension.

The rest of the paper is structured as follows. In Section 2 the motivations and requirements for the new SPARQL extension are provided. Section 3 provides the details of the extension. Section 4 discusses possible realization of the framework. Section 5 discusses the related work, and Section 6 concludes the paper.

2 Motivation and Requirements

Why GROUP BY is Not Sufficient?. In order to provide the motivation and discuss the requirements for the new SPARQL extension let us present the following motivating scenarios.

Scenario 1. The user Anna searches for natural monuments located in East-Central Europe. She would like to have the monuments grouped by their proximity.

Scenario 2. The user José submits a query on Portuguese paintings. He wants the answers to be grouped by movements (styles of artworks).

Classically, aggregation abilities are provided by SQL-like GROUP BY clause. Although GROUP BY is not officially supported by SPARQL, there are engines (for example Virtuoso[2], Jena[3]), that support the GROUP BY functionality. Despite that, after the analysis of the above scenarios, one can notice that classical GROUP BY semantics is not proper to solve either of the above tasks.

Let us consider the first scenario. Would be grouping by longitude and latitude what the user Anna actually expects? GROUP BY semantics is to partition the results by identical values. Since no natural monuments share the same coordinates, as an effect of such grouping one row for each value would be created, which obviously would not meet the needs of the user.

Let us further consider the second scenario. In this case it is possible that two paintings share the same style. However, by simply using GROUP BY the user José may still obtain too many groups to easily interpret them. For example the list of the art movements that belong to the *avant-garde* movement is as follows: *art nouveau*, *conceptual art*, *cubism*, *expressionism* and many more. Moreover the groups may be related to each other (e.g. by subclass-superclass relation) what is not captured by the semantics of the GROUP BY clause.

Since the semantics of GROUP BY is "crisp", in both discussed cases the user obtains too many results. Notice also that in case of scenarios involving distributed and open environments, another general problem of GROUP BY may be the requirement for a user to understand the data apriori - before submitting a query. While grouping could be considered as helpful for users to understand the data granularity, the semantics of GROUP BY actually requires from users this knowledge to specify good grouping conditions in a query.

[2] http://virtuoso.openlinksw.com

[3] http://jena.sourceforge.net/ARQ/

Since there are problems with achieving the desired behaviour in a declarative fashion by means of classical GROUP BY, the proposition to extend SPARQL with CLUSTER BY statement is justified.

Design Issues for CLUSTER BY. There are several issues that have been identified to address during the design of the functionality of the new SPARQL extension. These issues follow into two main categories:

1. What would be the output of a query with `CLUSTER BY` statement?
2. How to incorporate clustering algorithms and their parameters into the framework?

As pointed out in [14], a SPARQL query consists of three parts. The *pattern matching* part offers the features of graph pattern matching like optional parts, union of patterns, filtering etc., and the possibility of indicating the data source to be matched by a pattern. The computed output of the pattern may be modified by *solution modifiers*, which are classical operators such as projection, distinct, order etc. The third part of the query is constituted by its *output* which may be of different types such as yes/no queries, selections of values of variables matching the patterns, construction of new triples, or description of resources.

The desired semantics of the query enhanced with CLUSTER BY would be to provide the original query results itselves as well as the assigned clusters. It is also worth noting that some clustering algorithms are able to generate a hierarchy of clusters. Such hierarchy provides additional meaning to the generated clusters. It might be also desirable to provide an intensional description for each cluster to supply the user with the generalized information about the cluster content. These descriptions should be then retrieved as the result of a query as well.

Taking into account the above considerations, as an answer to the first question, the following important design choices for the CLUSTER BY extension are proposed:

– since clustering is supposed to operate on originally generated query results, we propose to integrate CLUSTER BY clause execution after the *pattern matching* part of a query, that is in its *solution modifiers* part,
– we propose to separate all the metadata concerning the generated clusters (e.g. cluster descriptions, cluster hierarchy) from the results of a query themselves. In order to achieve this we propose to add to each result one variable that would store the identifier of the cluster assigned to the result. Moreover additionally to the cluster idenfitiers in the query answers, we propose to generate clustering metadata as the new *output* type in the form of a set of triples.

To answer the second design question, one needs to provide a solution for incorporating the clustering algorithms execution to an effect of executing a query enhanced with CLUSTER BY. This incorporation should be seamless, that is any other steps should not be required to initiate the clustering process, except submitting a query itself. For this reason we propose to add additional syntax elements to SPARQL grammar that would allow the specification of the clustering parameters.

In order to realize the proposed design choices we need to extend SPARQL by the new elements and define semantics for them. In the next section we describe the details of these extensions.

3 Extending SPARQL with CLUSTER BY

3.1 Preliminary Definitions

Let I, B, and L be a pairwise disjoint sets of IRIs, blank nodes, and literals respectively. A triple $(s, p, o) \in (I \cup B) \times I \times (I \cup B \cup L)$ is called and *RDF triple*, where s is the *subject*, p the *predicate* and o the *object*. Let V be the infinite set of query variables disjoint from the above sets. A *basic graph pattern (BGP)* is a set of triple patterns which are RDF triples. An *RDF graph* is a set of RDF triples.

A SPARQL query evaluation is based on matching graph patterns to subgraphs in the queried RDF graphs. A graph matching algorithm for evaluating SPARQL query tries to find an exhaustive set of mappings $\mu(v \to t)$ of query variables $v \in V$ to RDF terms t, where t is a member of the set union $(I \cup B \cup L)$. A mapping μ is called a *solution mapping*. By Ω is denoted a multiset (or bag) of possible solution mappings.

3.2 Grammar of CLUSTER BY

One of the important goals of this work is to follow the original syntax and semantics of SPARQL as much as possible.

Table 1 shows an excerpt of the grammar of SPARQL SELECT rule extended with CLUSTER BY. The extensions and new rules are shown in bold.

The *ClusterByClause* statement is added to the standard SPARQL grammar rule of *SolutionModifier*. The *ClusterByClause* is further expanded, and a new keyword `CLUSTER BY` is introduced. After the `CLUSTER BY` keyword there should be specified a list of variables according to which the results are supposed to be clustered. Furthermore, a new keyword `AS` is introduced, after which there should be

Table 1. The excerpt of the extended SPARQL grammar

SelectQuery	::=	'SELECT' ('DISTINCT'|'REDUCED')? (**Cluster?** Var+|'*') DatasetClause* WhereClause SolutionModifier
SolutionModifier	::=	OrderClause? LimitOffsetClauses? **ClusterByClause?**
ClusterByClause	**::=**	**'CLUSTER' 'BY' Var+ 'AS' Cluster UsingClause?**
UsingClause	**::=**	**'USING' Method ('('Params')')?**
Cluster	::=	**VAR1**
Method	::=	**IRI_REF**
Params	::=	**TriplesBlock**

Table 2. Prefixes used in the paper

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dc: <http://purl.org/dc/elements/1.1/>
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
@prefix dbpedia: <http://dbpedia.org/resource/>
@prefix dbpedia-owl: <http://dbpedia.org/ontology/>
@prefix smo: <http://www.ifi.uzh.ch/ddis/sparql-ml/mining.owl#>
@prefix sqrco: <http://www.cs.put.poznan.pl/sqrco.owl#>
@prefix ex: <http://www.example.org#>
@prefix art: <http://www.art_movements.example.org#> .
```

specified a variable that would be bound to the identifier of the cluster. The same variable should be also listed in the variable list in the *SelectQuery* statement. To allow the specification of clustering parameters there is an optional *UsingClause* provided in the *ClusterByClause*. This clause can be omitted, which would indicate the query answering engine is supposed to use the default clustering method. The *UsingClause* adds a new keyword `USING` which is followed by an indentifier of a clustering method and optional list of the method parameters. As there are numerous clustering methods and algorithms available, in order to support flexible choice, we propose the parameters to be specified by a set of triples in which the terms from some common ontologies and vocabularies are used.

Table 3 shows two example queries that correspond to the motivating scenarios from Section 2 and use the proposed syntax. All the prefixes used in the paper are listed in Table 2.

3.3 Semantics of CLUSTER BY

In order to enable specifying the cluster associated with an answer to SPARQL query we define a notion of *clustered solution mappings*. This definition is based on the definition of solution mapping as proposed in the SPARQL specification [16].

Definition 1 (Clustered Solution Mapping). *A* clustered solution mapping *$\dot{\mu}$ is a pair (μ, c), where μ is a solution mapping (as defined in [16]), and c is a cluster identifier associated with μ.*

Furthermore, we propose a new algebra operator, called *cluster assignement operator*, that assigns cluster identifiers to solution mappings. To every solution mapping it adds a new variable binding which maps a literal representing cluster identifier to the specified variable.

Definition 2 (Cluster Assignement Operator). *Let Ω be a multiset of solution mappings, let v be a query variable which is not bound in any $\mu \in \Omega$, and let c denote the cluster identifier to which the variable v is bound. The result of an application of the* cluster assignement operator *is a multiset of clustered solution mappings $\dot{\Omega}$ and is defined as follows*

$$ClusterAssignement(v, \Omega) = \{\dot{\mu} | (\mu, c) \in \dot{\Omega} \wedge \dot{\mu} = \mu \cup (v \rightarrow c)\}$$

Table 3. Example queries in which the syntax of CLUSTER BY clause is used

Query 1

```
SELECT ?x ?latitude ?longitude ?c
WHERE
?x rdf:type ex:NaturalMonument
?x locatedIn ex:East_Central_Europe
?x geo:lat ?latitude
?x geo:long ?longitude
CLUSTER BY ?latitude ?longitude AS ?c
USING  <http://www.cs.put.poznan.pl/sqrco/ExampleMethod>
(
   <http://www.cs.put.poznan.pl/sqrco/ExampleMethod>
       sqrco:usesAlgorithm
   <http://www.cs.waikato.ac.nz/ml/weka/SimpleKMeans>
   <http://www.cs.waikato.ac.nz/ml/weka/SimpleKMeans> smo:hasParam ?p1
   ?p1 rdf:type sqrco:NumberOfClusters
   ?p1 owl:hasValue xsd:integer('4')
)
```

Query 2

```
SELECT ?x ?y ?c
WHERE
?x rdf:type art:Painting
?x dbpedia-owl:movement ?y
?z dc:authorOf ?x
?z dbpedia-owl:nationality dbpedia:Portugal
CLUSTER BY ?y AS ?c
USING <http://www.cs.put.poznan.pl/sqrco/ExampleMethod>
(  <http://www.cs.put.poznan.pl/sqrco/ExampleMethod>
      sqrco:usesFeatureExtractionMethod
   <http://www.cs.put.poznan.pl/sqrco/ExampleFeatureExtractionMethod>
)
```

In order to enable getting cluster metadata together with the clustered answers as a result of a query we define an additional operator, called *cluster metadata operator*. As an output this operator produces a cluster hierarchy and cluster descriptions which are defined below.

Definition 3 (Cluster Hierarchy). *A* cluster hierarchy $\mathcal{H}$ *is a partially ordered set* $\langle \mathcal{C}, \succeq \rangle$*, where each element* $c \in \mathcal{C}$ *is a cluster identifier, and* $\succeq$ *is a binary relation that imposes ordering on the elements of* $\mathcal{C}$*.*

Definition 4 (Cluster Intensional Description). *Let* $c \in \mathcal{C}$ *be a cluster identifier, and* $\dot{\Omega}_c$ *be a set of clustered solutions of the form* $\dot{\mu}_i = (\mu_i, c), i = 1, ..., n$*. A* cluster intensional description $d \in \mathcal{D}$ *gives a meaning of all elements* $\dot{\mu}_i \in \dot{\Omega}_c$ *by specifying the necessary conditions that any element belonging to the set* $\dot{\Omega}_c$ *meets.*

Finally the cluster metadata operator is defined.

Definition 5 (Cluster Metadata Operator). *Let Ω be a multiset of solution mappings. The result of an application of the* cluster metadata operator *consists of: a hierarchy $\mathcal{H}$ of cluster identifiers $c \in \mathcal{C}$, and a set $\mathcal{D}$ of cluster intensional descriptions d:*

$$ClusterMetadata(v, \Omega) = \mathcal{H} \cup \mathcal{D}$$

In general, it is allowed that $\mathcal{D} = \emptyset$.

4 Realizing the Framework

The proposed framework for extending SPARQL does not prescribe any precise means to represent clustering parameters or clustering metadata such as cluster hierarchy and cluster descriptions.

In order to implement the framework the ontology *SPARQL Query Results Clustering Ontology (SQRCO)* has been developed that enables representing input parameters for clustering and metadata of the obtained results. For representing the clustering parameters SQRCO uses the terms from *SPARQL Mining Ontology (SMO)* [8] ontology as well as it defines its own terms. Table 3 shows the usage of the terms from SQRCO (for instance NumberOfClusters) to represent input parameters of the clustering.

The ontology is designed also to represent metadata of the generated clusters such as cluster identifiers, their hierarchical structure, and their intensional descriptions. For this purpose SQRCO includes concepts like ClusterID, IntensionalDescription, and properties parentOf, hasIntensionalDescription. The relation parentOf imposes ordering on the set of cluster identifiers which results in a hierarchy of cluster identifiers rooted at the cluster identifier `root`. For an illustration of how the terms from SQRCO are used to represent the generated clustering metadata Table 4 is provided that contains the example metadata for the art movements scenario discussed overall the paper.

There are numerous possibilities of incorporating feature extraction techniques, clustering algorithms, as well as concept learning methods (for intensional descriptions generation) into the proposed framework. The properties of a particular solution, such as computational complexity, would depend on the methods and algorithms chosen to fill the framework. The detailed discussion of solutions is out of the scope of this paper.

Some steps towards realizing an instantiation of the query results clustering idea have been done in the recent work [7,10] by the author of this paper. The work concerned query results clustering in case of data represented in Web Ontology Language (OWL), and included an implementation in JAVA, using KAON2 reasoning engine[4] and Weka data mining software[5]. The preliminary tests proved the feasibility of the query results clustering idea and its instantiation especially in terms of the running time which is crucial in on-the-fly query

[4] http://kaon2.semanticweb.org
[5] http://www.cs.waikato.ac.nz/ml/weka

Table 4. Set of triples representing clustering metadata with use of the terms from SQRCO ontology

```
<http://www.example.org#>     a owl:Ontology .

:root   a sqrco:ClusterID;
        sqrco:parentOf :c1, :c3 .
:c1     a sqrco:ClusterID;
        sqrco:hasIntensionalDescription :desc1;
        sqrco:parentOf :c2 .
:c2     a sqrco:ClusterID;
        sqrco:hasIntensionalDescription :desc2 .
:c3     a sqrco:ClusterID .

:desc1  a _:bn1 .
:desc2  a _:bn2 .

_:bn1   a owl:Class;
        owl:intersectionOf  (
        [
               owl:onProperty art:characterizedBy;
               owl:someValuesFrom  [
                   a owl:Class;
                   owl:unionOf  (
                  art:DramaticLight
                  art:DramaticColour  ) ] ]
        [
                owl:hasValue "17th Century"^^xsd:string;
                owl:onProperty art:hasLinguisticPeriod ] )  .

_:bn2   rdf:type  art:Baroque .
```

results processing. This encouraged further work on the idea and its formalization provided in this paper, where the precise specification of the framework for invoking SPARQL query results clustering by extending the query language is proposed. Another solutions under the proposed framework are current or ongoing work.

5 Related Work

There are numerous extensions of SPARQL in the literature. These range from extensions for computing semantic associations [9], through extensions to process spatio-temporal data [15], data streams [2], and trust values [6].

However, to the best of our knowledge ours is the first work on extending SPARQL with clustering abilities. The SPARQL syntax does not even support the standard grouping, yet. However, some of the engines (already mentioned in the paper) implement GROUP BY clause functionality following the semantics known from SQL. The idea of grouping and aggregating SPARQL query results with the GROUP BY clause functionality different from that of SQL was investigated in [17].

This work has been inspired by the Web search results clustering engines, Clusty, and Carrot2. Recently there has been also an interest in clustering structured query results. However, there are very few works on this topic so far, and they concern clustering the results of SQL queries [11,18]. As such these works assume different language of database, as well as different language of queries. For example, the approach proposed in [11] is designed for numerical data, and the one proposed in [18] is designed for discrete spatial data. Nevertheless, introducing new constructs into queries whose semantics is to cluster query results has already been proposed in [11], and [18].

In [3,4] the methods for clustering the data represented in OWL (description logics) were proposed. Several approaches to clustering Semantic Web data were explored in [5]. However, those works do not assume the specific task of clustering query results that we address, nor they propose any extensions of SPARQL.

The closest work to ours is the proposition of extending SPARQL with data mining abilities [8]. In that work, however, the idea is to first execute a special type of query to build a model of the data (by using a training set of data), and at the second stage, apply another kind of query for prediction or classification tasks using the already built model. This solution differs from what we wanted to achieve in the type of tasks addresed as well as in the functionality of the approach.

6 Summary and Future Work

The contribution of the paper is the proposition of a declarative way for invoking dynamic clustering of the results of queries submitted over Semantic Web data. In particular, the paper proposes the extension of SPARQL by a new statement, CLUSTER BY, that supplies SPARQL wih clustering abilities. The SPARQL grammar for the extension is proposed, and its semantics discussed. Some steps towards realizing the proposed framework are reported.

The plan for the future work assumes work on the implementation of the clustering solutions that could be plugged into the proposed framework.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284(5), 34–43 (2001)
2. Bolles, A., Grawunder, M., Jacobi, J.: Streaming SPARQL - Extending SPARQL to Process Data Streams. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 448–462. Springer, Heidelberg (2008)
3. Fanizzi, N., d'Amato, C., Esposito, F.: Induction of Optimal Semi-distances for Individuals based on Feature Sets. In: Proc. of the DL 2007 Workshop (2007)

4. Fanizzi, N., d'Amato, C., Esposito, F.: Conceptual Clustering and Its Application to Concept Drift and Novelty Detection. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 318–332. Springer, Heidelberg (2008)
5. Grimnes, G.A., Edwards, P., Preece, A.D.: Instance Based Clustering of Semantic Web Resources. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 303–317. Springer, Heidelberg (2008)
6. Hartig, O.: Querying Trust in RDF Data with tSPARQL. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 5–20. Springer, Heidelberg (2009)
7. Jozefowska, J., Lawrynowicz, A., Lukaszewski, T.: Clustering results of conjunctive queries over knowledge bases in OWL. In: Proc. of ESWC 2009 poster session (2009)
8. Kiefer, C., Bernstein, A., Locher, A.: Adding Data Mining Support to SPARQL via Statistical Relational Learning Methods. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 478–492. Springer, Heidelberg (2008)
9. Kochut, K., Janik, M.: SPARQLeR: Extended Sparql for Semantic Association Discovery. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 145–159. Springer, Heidelberg (2007)
10. Lawrynowicz, A.: Grouping results of queries to ontological knowledge bases by conceptual clustering. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS (LNAI), vol. 5796, pp. 504–515. Springer, Heidelberg (2009)
11. Li, C., Wang, M., Lim, L., Wang, H., Chang, K.C.-C.: Supporting ranking and clustering as generalized order-by and group-by. In: Proc. of SIGMOD 2007, pp. 127–138 (2007)
12. Osinski, S., Weiss, D.: Carrot2: Design of a Flexible and Efficient Web Information Retrieval Framework. In: Szczepaniak, P.S., Kacprzyk, J., Niewiadomski, A. (eds.) AWIC 2005. LNCS (LNAI), vol. 3528, pp. 439–444. Springer, Heidelberg (2005)
13. Osinski, S., Weiss, D.: A Concept-Driven Algorithm for Clustering Search Results. IEEE Intelligent Systems 20(3), 48–54 (2005)
14. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and Complexity of SPARQL. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 30–43. Springer, Heidelberg (2006)
15. Perry, M., Sheth, A., Jain, P.: SPARQL–ST: Extending SPARQL to Support Spatiotemporal Queries, Kno.e.sis Center Technical Report. KNOESIS-TR-2009-01 (November 3, 2008), `http://knoesis.org/students/prateek/sparql-st-www09-tr.pdf`
16. Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF. Technical report, W3C Recommendation (January 15, 2008)
17. Seid, D.Y., Mehrotra, S.: Grouping and Aggregate queries Over Semantic Web Databases. In: Proc. of ICSC 2007, pp. 775–782. IEEE Computer Society, Los Alamitos (2007)
18. Zhang, C., Huang, Y.: Cluster By: a new sql extension for spatial data aggregation. In: Proc. of 15th ACM International Symposium on Geographic Information Systems, ACM-GIS 2007, p. 53 (2007)