Development of Faceted and Synonym Search Support for the Ontology Application Management Framework

Marut Buranarach[1], Pattama Krataithong[1,2], Nichanan Poovanavirote[3],
Palita Anantanitivate[3], and Nussara Siriset[3]

[1] Language and Semantic Technology Laboratory
National Electronics and Computer Technology Center (NECTEC), Pathumthani, Thailand
{marut.bur,pattama.kra}@nectec.or.th
[2] Department of Computer Science, Faculty of Science and Technology
Thammasat University, Pathumthani, Thailand
[3] Department of Computer Science, Faculty of Science, Kasetsart University, Bangkok, Thailand

Abstract. Semantic search is a form of search that goes beyond keyword-based searching. Searching based on keywords typically has several disadvantages including homonym and synonym problems which can reduce the retrieval effectiveness of a search system. Ontology-based search is a form of semantic search that can be applied to searching structured data, i.e. RDF data, which are exported from relational database. The Ontology Application Management (OAM) framework can provide support for ontology-based search application development over RDF data using an application template that generates queries based on SPARQL template. However, OAM still relies on keyword-based search when the properties are datatype properties, i.e. those having property values as literals. In this paper, we propose to use faceted and synonym search to augment the keyword-based search over datatype property values. One of the main goals is to provide a generic framework for improving the effectiveness of searching RDF data. Our system design adopted the service-oriented architecture (SOA) approach in creating reusable service components. Two key components are synonym and aggregation service Web APIs. The SPARQL query templates for implementing synonym and faceted search are described. Finally, we demonstrate an adoption of the framework in searching a large-scale database in the professional qualification domain.

Keywords: ontology-based search, semantic search, RDF, SPARQL aggregation

1 Introduction

Semantic search is a form of search that goes beyond keyword-based searching. Searching based on keywords typically has several disadvantages. First, there are many different terms that share the same meanings, i.e. synonyms. If the user's query terms do not match with the terms in the document or database, the retrieval effectiveness will be reduced. Second, one term can mean different things, i.e. homonyms. When the user's query terms are ambiguous, the retrieval effectiveness of the search system will be reduced. Thus, one goal of semantic search is to improve the retrieval effectiveness of traditional keyword-based searching.

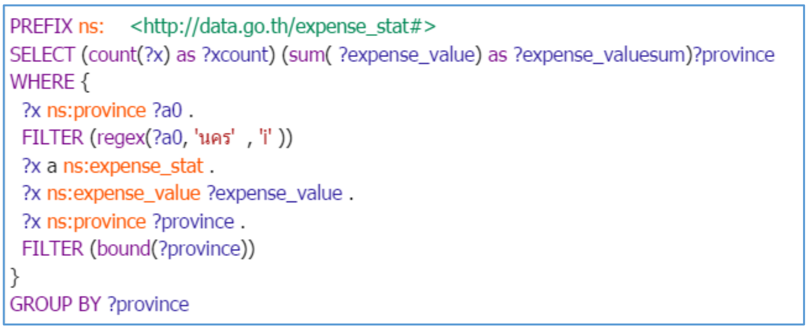
Ontology-based search is a form of semantic search that can be applied to searching structured data. Ontology in the OWL (Web Ontology Language) standard can be used to define structure of the RDF data exported from some database sources using some mapping languages, e.g. D2RQ mapping language [1], R2RML [2]. The resulted RDF data can be queried using SPARQL [3]. By applying ontology over RDF data, concept-based search can be conducted over the ontology-based inferenced data. The OAM Framework [4] is an application framework that can simplify ontology-based application development. The OAM framework can provide support for ontology-based search application development over RDF data using an application template that generates queries based on a SPARQL template.

OAM provides support for concept-based search based on property-value search conditions. Although this technique is effective when the properties in the search conditions are object properties, i.e. those having instances of some concepts as property values, it still relies on keyword-based search when the properties are datatype properties, i.e. those having property values as literals. In this paper, we propose to use faceted and synonym search to augment the keyword-based search over datatype property values. Synonym search using query expansion technique can improve coverage of retrieved resources when the search terms do not match with the database terms. Faceted browsing over the retrieved results can additionally group the results based on different properties and values which would allow the user to eliminate non-relevant results. In our framework, the service-oriented architecture (SOA) is adopted to allow for reusable components. The synonym and aggregation service Web APIs are among the key components of the framework. We describe some SPARQL query templates that enable synonym and faceted search support. Finally, we show our adoption of the framework in a search system over the Thailand Professional Qualification Database.

2 Background

2.1 SPARQL Aggregation Query

SPARQL 1.1 [3] provides support for aggregation operators. Aggregation functions allow the search results of multiple rows to be reduced into single values. Common aggregate functions include COUNT, COUNT DISTINCT, SUM, AVERAGE, MIN, and MAX. The results of aggregations can be partitioned into one or more groups based on the specified values in columns, i.e. GROUP BY. The partitioned results contain one row per aggregated group. The syntax of aggregations in SPARQL queries is similar to those of SQL aggregation queries. Fig. 1 shows an example of SPARQL aggregation query.



```
PREFIX ns:    <http://data.go.th/expense_stat#>
SELECT (count(?x) as ?xcount) (sum( ?expense_value) as ?expense_valuesum)?province
WHERE {
  ?x ns:province ?a0 .
  FILTER (regex(?a0, 'นคร'  , 'i' ))
  ?x a ns:expense_stat .
  ?x ns:expense_value ?expense_value .
  ?x ns:province ?province .
  FILTER (bound(?province))
}
GROUP BY ?province
```

Fig. 1. Example of SPARQL Aggregation Query

2.2 Concepts and Terms

From an ontology viewpoint, concepts are abstract representation of objects in the real-world. Concept meanings are independent of the terms used to refer to the concepts. A term may represent different concepts, i.e. homonyms. Different terms may also represent the same concept, i.e. synonyms. Fig. 2 shows the relationship between terms, concepts and objects.

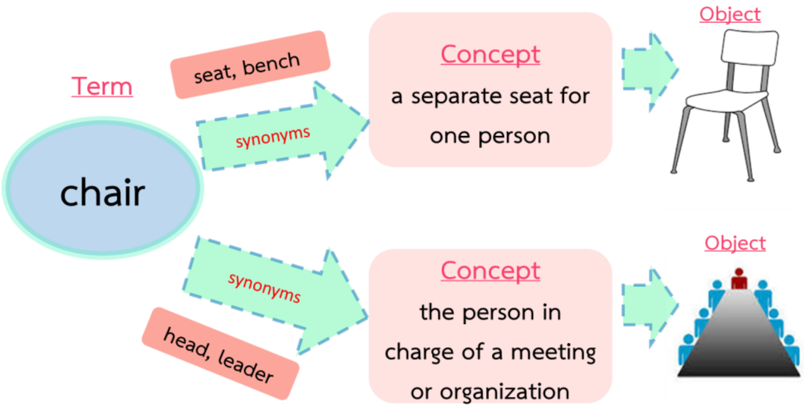

Object
Term
seat, bench
synonyms
Concept
a separate seat for one person
chair
Object
synonyms
head, leader
Concept
the person in charge of a meeting or organization

Fig. 2. Relationship between term, concept and object

The ambiguity of meanings of some terms can cause problems in searching. Specifically, a 'hyponym' or ambiguous search term can reduce precision of the retrieval while a search term that has synonyms can result in reduced recall. Thus, different concept-based searching techniques based on ontology are proposed to overcome limitation of keyword search [5, 6]. These techniques include query expansion and faceted search [7].

2.3 The Ontology Application Management (OAM) Framework

The Ontology Application Management (OAM) Framework [4] is a java-based web application development platform which helps user to build a semantic web application without programming skill required. The underlying technology of OAM is Apache Jena [8], D2RQ and RDF data storage. OAM includes three main modules follows as:

- Database-to-Ontology Mapping provides a user interface for mapping between an existing relational database schema and ontology file (OWL). This process helps users who do not have a programming skill in mapping and converting relational database data to RDF format.
- Ontology-based Search Engine provides a form-based SPARQL data querying service for users to query each dataset by defining search conditions.
- Recommendation Rule System provides a simplified interface for rule management. Users can define a condition of rules that do not require knowledge of the rule syntax of reasoning engine.

OAM has been used to support development of many ontology-based applications in different domains including smart home [9], excise duty [10] and open data [11] domains.

3 Faceted and Synonym Search Support Framework for OAM

3.1 System Architecture

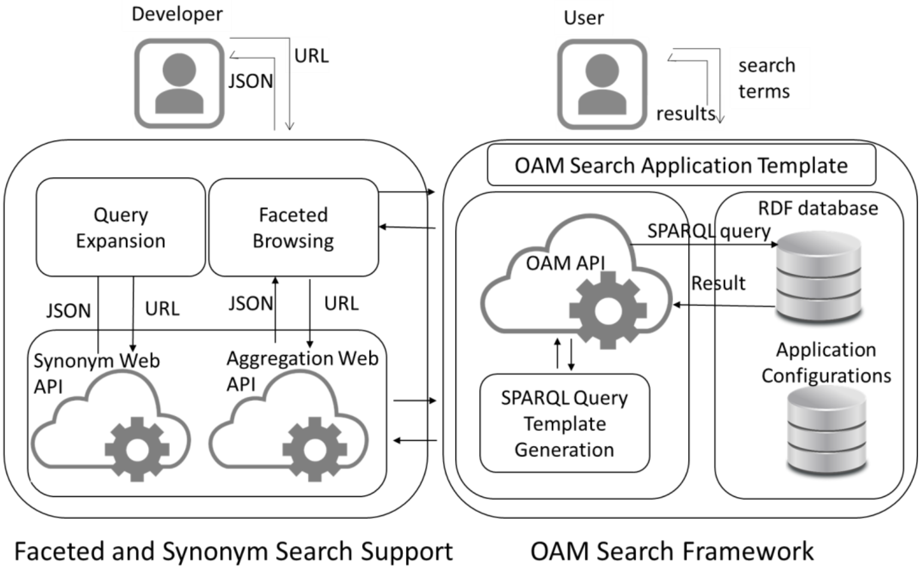


Fig. 3. System architecture of the faceted and synonym search support component of OAM

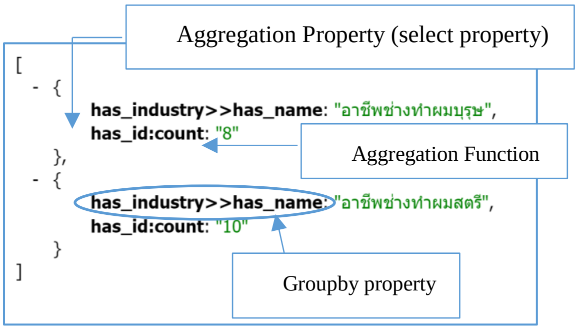
The faceted and synonym search support component is developed using the service-oriented architecture (SOA). Fig. 3 shows an overall architecture of the system. The components consist of two main RESTFul Web APIs: synonym service and aggregation service APIs. The synonym service API is used by the query expansion module, which generates an expanded set of the user query terms. The aggregation service API is used by the faceted browsing module, which summarizes the search results according to different property values and their unique counts. Both components will interact with the SPARQL query template generation module to generate the SPARQL queries accordingly.

The faceted and synonym search support component is reusable and can be used by the OAM framework or other applications. The end-users can use the searching functions by means of the OAM search application template. Using the application template, the user can define search conditions in a search form which will be transformed to SPARQL queries in searching the RDF data. The query expansion technique will be applied to the user query terms for synonym-based search. After the results are retrieved, the faceted browsing module will list groups of the results by property values of different properties and unique counts of their members. When the user clicks on label of a group, the system adds a filter condition to the original search conditions of the user query to filter the search results. The synonym and aggregation service Web API may also be used independently of the OAM search application.

3.2 OAM Aggregation Web API

The OAM aggregation Web API applies aggregation functions on top of the OAM search Web API [4]. The OAM search Web API accepts the following parameters: dataset name, search properties, search operators and search values. The aggregation Web API adds the following parameters to the search Web API: aggregation function, aggregation property and groupby property. Fig. 4 shows an example of the aggregation API results in JSON format. In this example, searching in qualification database applied an aggregation function to count the number of search results based on a property of industry name. The parameters for the Web API are:

- *Aggregation function : count*
- *Aggregation property : has_id*
- *Groupby property : has_industry>>has_name*



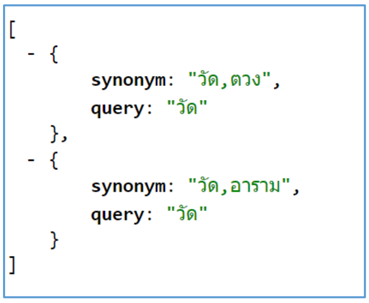
```
[
  - {
      has_industry>>has_name: "อาชีพช่างทำผมบุรุษ",
      has_id:count: "8"
    },
  - {
      has_industry>>has_name: "อาชีพช่างทำผมสตรี",
      has_id:count: "10"
    }
]
```

Fig. 4. Example of results of OAM aggregation Web API in JSON format

The results of unique property values and counts of their members are then returned. The results will be used to further refine the filter condition of the original query. Specifically, the returned group labels which represent each property value will be used as a filter condition added to the original query.

3.3 Synonym Service Web API and Synonym Management System

The synonym service Web API retrieves the synonym sets from the synonym database given a query term. Fig. 5 shows an example of the API results in JSON format. In this example, a query term ‘วัด’ is a homonym which can have the meaning of ‘measurement’ or ‘temple’. In this case, the synonym service API returns two synonym sets (‘วัด’,‘ตวง’) (‘วัด’,‘อาราม’) for the two different meanings accordingly.



```
[
  - {
        synonym: "วัด,ตวง",
        query: "วัด"
    },
  - {
        synonym: "วัด,อาราม",
        query: "วัด"
    }
]
```

Fig. 5. Example of results of the synonym service Web API in JSON format for a hyponym

A synonym management system was created to support domain experts in management of synonym sets. The system consists of a keyword extraction system, which selects some key terms from a database and list them as the seed words for the domain experts to define synonym sets. The system also provides synonym set merging function to allow the experts to group multiple synonym sets into one set. The system also provides a function of linking synonym set to a concept URI in ontology file.

3.4 SPARQL Query Templates

3.4.1 SPARQL Query Template for Faceted Search

```
PREFIX  ns:   <http://example.org/owl/onto1.owl#>
PREFIX  rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT  (count(?agg_prop) AS ?agg_prop_count) ?grpby_prop
WHERE
    { ?x rdf:type ns:Class1 .
      ?x ns:agg_prop ?agg_prop .
      ?x ns:grpby_prop ?grpby_prop .
      ?x ns:search_prop ?search_prop
      FILTER regex(?search_prop, "term1", "i")
    }
GROUP BY ?grpby_prop
```

Fig. 6. Example of SPARQL query template for listing values of a facet and their counts

In order to implement a faceted search, SPARQL aggregation queries must be formed in order to generate list of facets and its values. Fig. 6 shows an example of a SPARQL query template for listing values of a facet (‘ns:grpby_prop’) and their counts (counting values of ‘ns:agg_prop’) given a search conditions (e.g., ‘ns:search_prop’ contains “term1”). When the user click on a facet value, a filter condition is added to the original search condition as shown in Fig. 7.

```
FILTER (regex(?search_prop, "term1", "i") && ?grpby_prop = "-user_selected_value-")
```

Fig. 7. Modified FILTER condition of a query for faceted browsing when a value is selected

3.4.2 SPARQL Query Template for Synonym Search

```
PREFIX  ns:   <http://example.org/owl/onto1.owl#>
PREFIX  xsd:  <http://www.w3.org/2001/XMLSchema#>
PREFIX  rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT DISTINCT ?id
WHERE
  {  { ?x ns:search_prop1 ?a0
       FILTER ( ( regex(?a0, "syn1", "i") || regex(?a0, "syn2", "i") ) || regex(?a0, "syn3", "i") )
       ?x ns:id_prop ?id .
       ?x rdf:type ns:Class1
     }
    UNION
     { ?x ns:search_prop2 ?a1
       FILTER ( ( regex(?a1, "syn1", "i") || regex(?a1, "syn2", "i") ) || regex(?a1, "syn3", "i") )
       ?x ns:id_prop ?id .
       ?x rdf:type ns:Class1
     }  }
```

Fig. 8. Example of SPARQL query template for synonym-based search

In order to implement a synonym search, a SPARQL query combining query expansion must be formed in order to include the related synonym sets in the query. Fig. 8 shows an example of a SPARQL query template for creating a synonym search given a keyword as a search term over all the properties (‘ns:search_prop1’, ‘ns:search_prop2’). In this example, the synonym set for the search term consists of the following terms: “syn1”, “syn2”, and “syn3”. The results are a list of ID values (‘ns:ip_prop’) of the matched resource.

4 System Implementation and Scenario

The framework has been adopted in a search system for the Thailand Professional Qualification Institute (TPQI)’s TPQI-Net database[1]. The search system was built

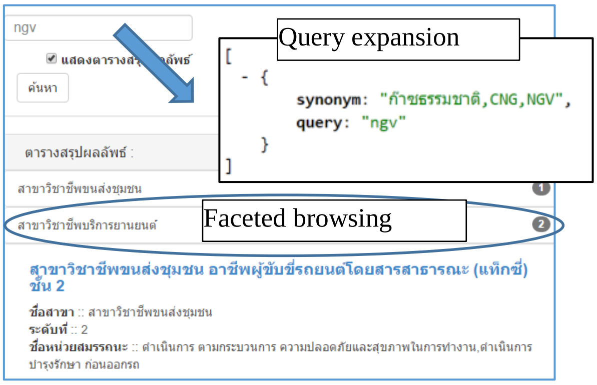
[1] http://tpqi-net.tpqi.go.th/tpqi_sa/

using the OAM framework. As of April 2016, the database contains the information of thousands of units of competencies of over 200 qualifications developed for over 25 industries.

One of the problems in searching the database was that the user's search terms are frequently informal terms that do not match with the technical terms used in the database. The synonym search approach was adopted in order to allow the user to use informal terms to search the qualification database. In addition, the faceted search approach was adopted to allow the users to filter the results according to the facets that they are interested in to improve the result accuracy.

The development of synonyms database starts with the keyword extraction service, which extracts the technical terms appeared in the standards database. The human experts then define some casual terms, which are those likely used as the user's query terms, as synonyms of each term. This results in the synonym sets database.

In processing a query, the search system uses the query expansion technique, which maps the user's query terms with the related synonym sets via the synonym service Web API. If matched, the user's query terms are then extracted to the database terms which will result in the retrieval of the related qualifications. Using this technique, when the user's query term does not appear in the database but is defined in a synonym set, the search results will be equivalent to those of the keyword-based search using the associated term appeared in the database.



```
[
  - {
        synonym: "ก๊าซธรรมชาติ,CNG,NGV",
        query: "ngv"
    }
]
```

สาขาวิชาชีพขนส่งชุมชน อาชีพผู้ขับขี่รถยนต์โดยสารสาธารณะ (แท็กซี่) ชั้น 2

ชื่อสาขา :: สาขาวิชาชีพขนส่งชุมชน
ระดับที่ :: 2
ชื่อหน่วยสมรรถนะ :: ดำเนินการ ตามกระบวนการ ความปลอดภัยและสุขภาพในการทำงาน,ดำเนินการ บำรุงรักษา ก่อนออกรถ

Fig. 9. Example of synonym search combined with faceted browsing in TPQI-Net system

Fig. 9 shows an example search results when using synonym-based search in combination with faceted browsing. In this example, the user uses a query term "NGV" which does not appear in the database. When the synonym search is applied, the related database terms included in its synonym set will be used. Thus the relevant results will be returned. The user can further filter the results based on a facet of industry name, which subsequently improves the accuracy of the search results.

5 Conclusion

In this paper, we describe an approach to extend the OAM framework to support faceted and synonym search over RDF data. One of the main goals is to provide a generic framework for improving the effectiveness of searching RDF data. Our system design adopted the SOA approach in creating reusable service components. Two key components are synonym and aggregation service Web APIs. The SPARQL query templates for implementing synonym and faceted search are described. Finally, we demonstrate an adoption of the framework in searching a large-scale database in the professional qualification domain. Our future work will focus on supporting faceted browsing of hierarchical structure of property values and extending the synonym service to support more relationship types of terms, e.g. is-a and part-of relations.

References

1. Bizer, C., Seaborne, A.: D2RQ-Treating Non-RDF Databases as Virtual RDF Graphs. In: Poster at the 3rd International Semantic Web Conference (ISWC2004) (2004).
2. W3C: R2RML: RDB to RDF Mapping Language, https://www.w3.org/TR/r2rml/.
3. W3C: SPARQL 1.1 Query Language, https://www.w3.org/TR/sparql11-query/.
4. Buranarach, M. et al.: OAM: An Ontology Application Management Framework for Simplifying Ontology-Based Semantic Web Application Development. Int. J. Softw. Eng. Knowl. Eng. 26, 01, 115–145 (2016).
5. Mäkelä, E.: Survey of semantic search research. In: Proceedings of the seminar on knowledge management on the semantic web (2005).
6. Mangold, C.: A Survey and Classification of Semantic Search Approaches. Int. J. Metadata Semant. Ontologies. 2, 23–34 (2007).
7. Hearst, M.: Design recommendations for hierarchical faceted search interfaces. In: ACM SIGIR Workshop on Faceted Search (2006).
8. Carroll, J.J., Reynolds, D., Dickinson, I., Seaborne, A., Dollin, C., Wilkinson, K.: Jena: Implementing the Semantic Web Recommendations. In: Proc. of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, pp. 74–83 (2004).
9. Wongpatikaseree, K., Ikeda, M., Buranarach, M., Supnithi, T., Lim, A. O., and Tan Y.: Activity Recognition using Context-Aware Infrastructure Ontology in Smart Home Domain. In: Proc. of the 7th International Conference on Knowledge, Information and Creativity Support Systems (KICSS2012) (2012).
10. Buranarach, M., Ruangrajitpakorn, T., Anutariya, C., Wuwongse, V.: Ontology Design Approaches for Development of an Excise Duty Recommender System. In: Kawtrakul, A., Laurent, D., Spyratos, N., and Tanaka, Y. (eds.) Information Search, Integration, and Personalization. CCIS, vol. 421, pp. 119–127. Springer International Publishing (2014).
11. Krataithong, P., Buranarach, M., and Supnithi, T.: RDF Dataset Management Framework for Data.go.th. In: Proc. of the 10th International Conference on Knowledge, Information and Creativity Support Systems (KICSS2015) (2015).