Falcons: Searching and Browsing Entities on the Semantic Web

Gong Cheng
gcheng@seu.edu.cn

Weiyi Ge
wyge@seu.edu.cn

Yuzhong Qu
yzqu@seu.edu.cn

Institute of Web Science, School of Computer Science and Engineering
Southeast University, Nanjing 210096, P.R. China

ABSTRACT

As of today, the amount of data on the Semantic Web has grown considerably. The services for searching and browsing entities on the Semantic Web are in demand. To provide such services, we developed the Falcons system. In this poster, we present the features of the Falcons system.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Design, Experimentation

Keywords

Indexing, Search Engine, Semantic Web, Summarization

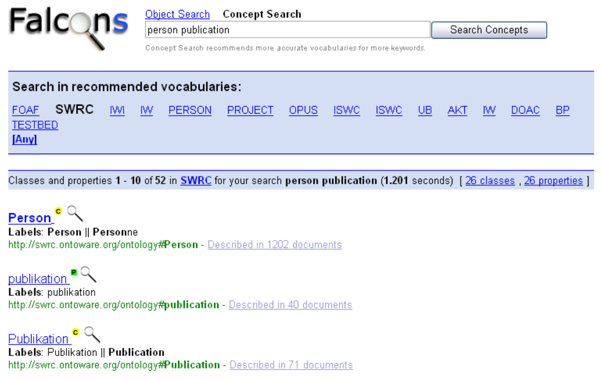
1. INTRODUCTION

More and more RDF data have been published on the Semantic Web, and searching for entities (concepts and objects) on the Semantic Web is in demand. To serve it, we developed the Falcons[1] system. At the time of writing, more than 7 million well-formed RDF documents, containing 250 million RDF statements, have been discovered by Falcons, and 4,400 ontologies have been identified among them. About 30 million Semantic Web entities have been indexed, and about 2 million of them are concepts (classes or properties). This poster presents the services provided by the Falcons system and its supporting technical features.

2. SYSTEM FUNCTIONALITY

Falcons provides keyword-based search for Semantic Web entities. In the search results page, for each entity, its types and labels are presented for users to quickly understand its denotation. We also present the number of RDF documents where each entity is used, to show its popularity. We associate each entity with a link to the page listing the RDF documents that define and use it.

[1] `http://iws.seu.edu.cn/services/falcons/`.



person publication

Concept Search recommends more accurate vocabularies for more keywords.

Search in recommended vocabularies:

FOAF **SWRC** IWI IW PERSON PROJECT OPUS ISWC ISWC UB AKT IW DOAC BP TESTBED
[Any]

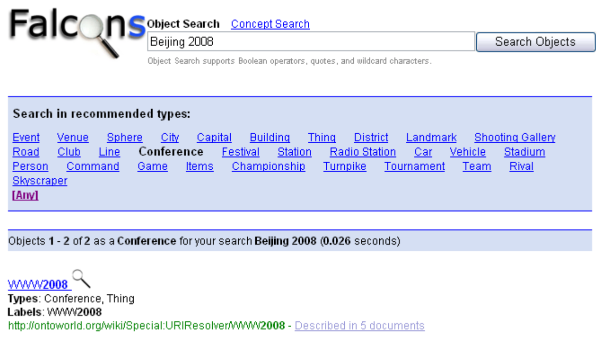
Classes and properties **1 - 10** of **52** in **SWRC** for your search **person publication** (**1.201** seconds) [26 classes , 26 properties]

Person
Labels: **Person** || **Person**ne
http://swrc.ontoware.org/ontology#**Person** - Described in 1202 documents

publikation
Labels: publikation
http://swrc.ontoware.org/ontology#**publication** - Described in 40 documents

Publikation
Labels: Publikation || **Publication**
http://swrc.ontoware.org/ontology#**Publication** - Described in 71 documents

Figure 1: A screenshot of Falcons Concept Search.



Beijing 2008

Object Search supports Boolean operators, quotes, and wildcard characters.

Search in recommended types:

Event Venue Sphere City Capital Building Thing District Landmark Shooting Gallery Road Club Line **Conference** Festival Station Radio Station Car Vehicle Stadium Person Command Game Items Championship Turnpike Tournament Team Rival Skyscraper
[Any]

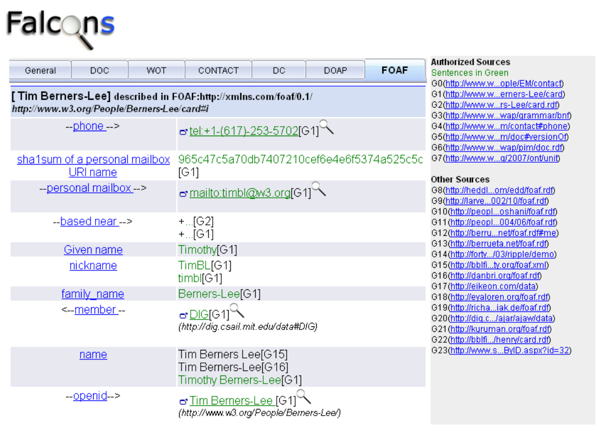
Objects **1 - 2** of **2** as a **Conference** for your search **Beijing 2008** (**0.026** seconds)

WWW**2008**
Types: Conference, Thing
Labels: WWW**2008**
http://ontoworld.org/wiki/Special:URIResolver/WWW**2008** - Described in 5 documents

Figure 2: A screenshot of Falcons Object Search.

Semantic Web developers need to search existing ontologies for reuse in their data. To serve it, Falcons not only presents the classes and properties that match the query terms, but also dynamically recommends ontologies. Once an ontology is selected, as depicted in Fig. 1, the results will be refined to only include the classes and properties in that ontology. Such interaction mode enables users to not only understand specific classes and properties but also obtain a general view of ontologies.

Semantic Web developers and ordinary users also need to search for the objects on the Semantic Web. To serve it, Falcons presents the objects that match the query terms, and also dynamically recommends several types of objects that the user is probably searching for. Once a type is se-



[Tim Berners-Lee] described in FOAF:http://xmlns.com/foaf/0.1/
http://www.w3.org/People/Berners-Lee/card#i

--phone -->	tel:+1-(617)-253-5702[G1]
sha1sum of a personal mailbox URI name	965c47c5a70db7407210cef6e4e6f5374a525c5c [G1]
--personal mailbox -->	mailto:timbl@w3.org[G1]
--based near -->	+...[G2] +...[G1]
Given name	Timothy[G1]
nickname	TimBL[G1] timbl[G1]
family_name	Berners-Lee[G1]
<--member --	DIG[G1] *(http://dig.csail.mit.edu/data#DIG)*
name	Tim Berners Lee[G15] Tim Berners-Lee[G16] Timothy Berners-Lee[G1]
--openid-->	Tim Berners-Lee [G1] *(http://www.w3.org/People/Berners-Lee/)*

Authorized Sources
Sentences in Green
G0(http://www.w...ople/EM/contact)
G1(http://www.w...erners-Lee/card)
G2(http://www.w...rs-Lee/card.rdf)
G3(http://www.w...wap/grammar/bnf)
G4(http://www.w...m/contact#phone)
G5(http://www.w...m/doc#versionOf)
G6(http://www.w...wap/pim/doc.rdf)
G7(http://www.w...g/2007/ont/unit)

Other Sources
G8(http://heddl...om/edd/foaf.rdf)
G9(http://larve...002/10/foaf.rdf)
G10(http://peopl...oshani/foaf.rdf)
G11(http://peopl...004/06/foaf.rdf)
G12(http://berru...net/foaf.rdf#me)
G13(http://berrueta.net/foaf.rdf)
G14(http://forty.../03/ripple/demo)
G15(http://bblfi...ty.org/foaf.xml)
G16(http://danbri.org/foaf.rdf)
G17(http://eikeon.com/data)
G18(http://eyaloren.org/foaf.rdf)
G19(http://richa...iak.de/foaf.rdf)
G20(http://dig.c.../ajar/ajaw/data)
G21(http://kuruman.org/foaf.rdf)
G22(http://bblfi.../henry/card.rdf)
G23(http://www.s...ByID.aspx?id=32)

Figure 3: A screenshot of Falcons Entity Summary.

lected, as depicted in Fig. 2, the results will be refined to only include the matched objects of that type. So it can help users quickly find the desired objects.

For each entity, to help users quickly understand it, Falcons extracts a set of RDF statements about it from various data sources on the Semantic Web and organizes them into a summary. As depicted in Fig. 3, these statements are not simply listed, but are clustered by different ontologies such as FOAF and Dublin Core (DC) according to their predicates. Users can switch between the tabs to browse the statements described by using a specific ontology, which often characterize a specific aspect of an entity. In addition, each RDF statement is associated with its source.

3. TECHNICAL FEATURES

3.1 Finding Entities by Virtual Documents

We use information retrieval (IR) techniques to index entities. For each entity, we index all the terms from its virtual document [2]. The virtual document of an entity consists of its names (local name and labels), other associated literals, and the names of its neighboring entities in RDF graphs, decoded from all the RDF documents on the Semantic Web. All these terms become effective especially when the query terms do not mention the names of the desired entities. For the example depicted in Fig. 2, actually the labels of `ontoworld:WWW2008`[2] do not contain the query term "Beijing" but Falcons can still find it because the value of its "has-location-city" property is `ontoworld:Beijing`.

In this way, each term is indexed to more entities than in the traditional methods. So we devise a weighting scheme to ensure that well matched entities, e.g., whose labels match the query terms, will be ranked higher. In ranking, the popularity of entities is also considered.

3.2 Recommending Ontologies for Concept Search

For each ontology, we index the virtual documents of all its classes and properties. So for each query, the candidate ontologies can be immediately obtained. The recommendation is based on a combination of the TF-IDF technique and the popularity of ontologies. The ontologies that contain widely instantiated classes and properties will be more likely to be recommended.

We also index each ontology to its classes and properties. So, combined with the inverted index from terms to concepts, users can be served with only those matched concepts in a specific ontology.

3.3 Recommending Classes for Object Search

To obtain the candidate classes for recommendation, it is impractical to directly build an index from terms to classes because some classes may have too many instances as well as too many terms to be indexed. Instead, we iterate over the search results on the fly to collect the classes of the resulting objects stored in the index. Then, these classes are ranked based on the coverage of their instances in the results. The top-ranked classes are selected, dynamically grouped by their names, and recommended to users.

We also index each class to its instances. So, combined with the inverted index from terms to objects, users can be served with only those matched objects of a specific type.

At the time of writing, the object search service has just been enhanced to allow users to navigate class hierarchies for query restriction. This new feature is enabled by a way of class subsumption reasoning on multiple vocabularies and an improved technique for recommending classes.

3.4 Summarizing Entities for Browsing

RDF statements from different data sources are stored and indexed. For each entity, a set of statements about it is extracted and ranked according to the popularity of the entities in these statements. The MMR technique [1] is used to rerank these statements, i.e., the statements are selected into the summary one by one, and once a statement is selected, the ranking values of the remaining statements with a similar predicate (from the same ontology) will be decreased. Such method can improve the diversity of the summaries.

4. CONCLUSION AND FUTURE WORK

Falcons is a keyword-based search system for concepts and objects on the Semantic Web, and is equipped with entity summarization for browsing. Future work includes improving the manipulation of concept spaces for a better user experience in searching and browsing the Semantic Web.

5. ACKNOWLEDGMENTS

This work is supported by the NSFC under Grant 60773106. We are also grateful to Honghan Wu and Xiang Zhang for their work on the system.

6. REFERENCES

[1] Carbonell, J. and Goldstein, J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. SIGIR*, pages 335–336, 1998.

[2] Qu, Y., Hu, W., and Cheng, G. Constructing virtual documents for ontology matching. In *Proc. WWW*, pages 23–31, 2006.

[2]The prefix `ontoworld` indicates `http://ontoworld.org/wiki/Special:URIResolver/`.