Title	The ExpertFinder Corpus 2007 for the Benchmarking and Development of ExpertFinding Systems
Author(s)	Hogan, Aidan; Harth, Andreas
Publication Date	2007
Publication Information	Aidan Hogan, Andreas Harth "The ExpertFinder Corpus 2007 for the Benchmarking and Development of ExpertFinding Systems", Proceedings of the 1st International ExpertFinder Workshop, 2007.
Item record	http://hdl.handle.net/10379/564

schema and instance integration is possible for these sources of data in Section 3. We then proceed to analyse the various properties and possibilities of this dataset in Section 4. Section 6 concludes.

2. DATA ACQUISITION

In this section we briefly describe the raw data we have acquired and used to create the final dataset. We maintain that the DBLP and CiteSeer datasets, which relate to computer science publications, would be ideal for expert finding analyses using publication and author metadata. We also maintain that expert finding is also possible through analysis of social networks, and thus we also leverage FOAF data in the creation of our benchmark dataset.

These datasets (or more specifically their intersection) are confined to the computer science domain. The DBLP and CiteSeer datasets are available to download as compressed archives and are in XML and OAI formats respectively. We outline our treatment of these datasets in the next section.

The FOAF dataset is retrieved from the web using MultiCrawler [4] and is in RDF format. We specifically extract `foaf:Person` instances from the indexed data using the following query.
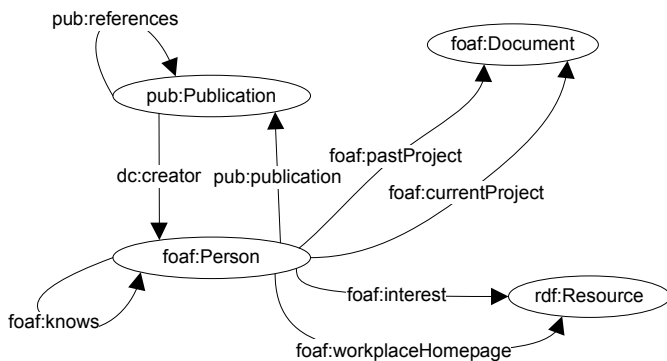
```
CONSTRUCT ?s ?p ?o .
 WHERE
 ?s rdf:type foaf:Person .
 ?s ?p ?o .
```

3. DATASET CREATION

This section outlines our approach to creating a unified dataset from the raw DBLP, CiteSeer and FOAF datasets. More generally, we show how multiple sources can be exploited and integrated to form one large well-linked integrated graph suitable for expert finding analyses. We do this under two distinct subheadings, one for explaining schema integration and another for outlining our approach to instance integration.

3.1 Schema Integration

The RDF data model has proven to lend itself well to excellent data integration. By converting to and creating data in RDF, similar concepts (i.e. entity types and their properties and relationships) from different native schemas can be mapped to one concept in RDF. CiteSeer and DBLP are both computer science publication datasets and thus, similar concepts are found in both (papers, authors, title,

1. INTRODUCTION

There has in recent years been a trend towards publishing data in structured formats, be it small datasets published by many groups of users or large sites publishing large data repositories. In the former category, we see an explosion in the use of XML formats such as RSS 2.0 and Podcasts, and of RDF formats such as FOAF, DOAP, SIOC, etc. In the latter category, we see sites such as CiteSeer, DBLP, WikiPedia, iMDB, US Patents etc. providing dumps of their databases in a structured formats under open licenses.

Structured datasets often have inherent graphs present, comprising of resources or instances as nodes and properties or predicates as edges. There has been some work done into analysing such graphs for prominent nodes [3] [6] [5] borrowing from more traditional work done in PageRank [1] and HITS [7]. Such prominent nodes which are representative of people can be interpreted as being experts.

Such structured datasets have their own native formats, schemas and topics. However, there exists a significant overlap between the datasets with regards to concepts and instances. To illustrate, observe that almost every dataset introduces the concept of a person, be that person an author, actor, friend etc, i.e. that concept overlap exists. Also observe that the same person may be described under FOAF, CiteSeer, DBLP etc. Thus we have schema overlap and instance overlap. In order to leverage data from multiple sources we require that the data be integrated with regards instance and schema. Without integration, we would observe multiple isolated data graphs; with integration we observe one large graph suitable for analysis.

In this paper we analyse a combined graph stemming from the FOAF, CiteSeer and DBLP datasets. We explain how we acquired this data in Section 2. We show how both

Figure 1: Main `owl:ObjectProperties` in the dataset, including domain and range. Other ontologies provide more `owl:ObjectProperties` and also `owl:DatatypeProperties`.

abstract etc.). By using the same schema for conversion (i.e. recycling equivalent concepts), we can achieve "schema integration" of the two sources of data. With this in mind, we retrieve and convert CiteSeer and DBLP to RDF using the same target schema, which we call the Research Publications Ontology [1].

This schema uses the local prefix `pub`, with a core concept of `pub:Pulication`. The schema fills in the gaps left by the Dublin Core and FOAF elements for describing publications and defines multiple subclasses of `pub:Publication` including `pub:InProceedings pub:MasterThesis pub:Book` etc. Essentially, the schema provides mappings from the useful concepts in DBLP and CiteSeer.

Figure 1 illustrates the typical relationships that exist between classes in the combined DBLP, CiteSeer and FOAF schemas.

3.2 Instance Integration

Now we have a dataset with equivalent concepts merged, but on the instance level, equivalent entries still exist. Therefore we apply "object consolidation" to achieve instance integration over the datasets. The result we have in mind is the merging of instances of the same papers from both CiteSeer and DBLP and the merging of equivalent person instances from the DBLP, CiteSeer and FOAF datasets. Essentially this involves identification of equivalents and merging their idenifiers to one consoldiated identifier.

Object consolidation involves analysis of properties defined as `owl:inverseFunctionalProperty`, and their values. The value of an inverse functional property is unique to a particular resource. Examples of such properties are `foaf:mbox`, `foaf:mbox_sha1sum` and `foaf:homepage`. For merging publications we use the `pub:ee` or electronic edition URI, which uniquely identifies the publication and thus is defined as an inverse functional property. If two instances have the same value for an inverse functional property, they are, by definition, equivalent.

A property is defined as being inverse functional in its respective ontology. In order to obtain a list of such inverse functional properties we visit the ontologies relevant to the dataset. The locations of these ontologies are assumed to be

the same as the namespace used by the classes and properties in the data.

With a list of inverse functional properties in hand, we can begin the object consolidation process, which works as follows:

- We scan the dataset to obtain a list of nodes which are equivalent. Equivalent instances are those which are bound together by the same value for the same inverse functional property.

- We then pick pivot identifiers which are the new identifiers assigned to the consolidated objects. URIs are chosen over blank node identifiers, and subceeding this rule, the most frequently used identifier is picked.

- The index is then rewritten, replacing old identifiers in the subject and object position with their respective pivot identifiers.

In theory, the process is iterative. If the object of an inverse functional property is changed, another iteration is required to ensure complete consolidation. However, in practice, such an occurance is rare. Values of inverse functional properties are generally quite static and rarely appear as a subject in a dataset.

We do not achieve full instance integration as many instances do not define any values of inverse functional properties. For example, a significant number of publication instances do not have a `pub:ee` property defined. It is possible to achieve more complete object consolidation by manually including "nearly inverse functional properties" or properties whose values would rarely be the same for different resources. An example would be `foaf:name`, which is not defined as being inverse functional, but which would usually be unique to a person in the dataset (i.e. authors often use middle initials etc. to distinguish themselves from their namesakes). We do not currently use this method in the creation of our testbed dataset.

4. PROPERTIES OF THE CORPUS

The resulting corpus is ~19GB in N-Triples format, and ~1.2GB gzipped. Tables 1 and 2 are the details of some of the characteristics of the corpus:

The corpus is available in both NQuads and RDF-NTriples [2]. NQuads is an extension of NTriples with the addition of a fourth element to the subject, predicate, object model; namely context which is the source of data. Some work has been done on including the context graph in links-analysis such as in [5].

The following is a list of possible queries that we foresee the corpus covering:

Characteristic	Value
Number of statements (quads/triples)	96,407,141
Number of instances	18,478,145
Number of distinct classes	224
Number of distinct predicates	719

Table 1: Breakdown on the count of instances, classes, predicates and statements.

Class	Count
`foaf:Person`	17,910,795
`pub:Publication`	715,690
`pub:InProceedings`	441,271
`pub:Article`	258,777
`foaf:Document`	55,582
`pub:Proceedings`	7,215
`pub:Incollection`	2,339
`pub:Book`	1,081

Table 2: Number of instances of most significant classes.

- find Semantic Web expert who is based near Dublin

- find expert in adapting Logic Programming techniques to the Web in my workplace

- find expert on Social Network Analysis which is close in my own social network i.e. that I may know

To give a more concrete example, the following is a list of requirements for a advertised position extracted from the DBWorld mailing list[3].

- A recent PhD degree in Computer Science or a related discipline.

- A strong track record of research and publications in the areas of multimedia (text, image, audio, video) recognition and analysis.

- Skilled in programming and experience in developing application prototypes.

In the following, we briefly discuss how a matching for advertised positions could be carried out. The first requirement can be partially matched by refining the search to `foaf:Person` instances with value `Dr` for property `foaf:title`. Publications of the candidate can be analysed from the CiteSeer or DBLP metadata for the keywords `multimedia recognition analaysis`. Values of `foaf:interest`, `foaf:currentProject` and also of `foaf:pastProject` etc. can be analysed to garner information on programming experience.

5. RELATED WORK

The initial version of DBLP in RDF was made available in 2004[4]. There is a substantial list of related vocabularies. Related vocabularies are Dublin Core[5], the KnowledgeWeb Portal ontology[6], Karlsruhe's SWRC Ontology [8], the AKT Portal Ontology [7], and the Semantic Web Portal Ontology [2].

The SwetoDblp ontology[8] enriches an RDF representation of DBLP with university information, in size comparable with our dataset. In addition of using DBLP as well as Citeseer and Web data, we include queries in form of job postings in our dataset.

TREC[9] has an "expert search task" as part of the Enterprise Track. However, their dataset is tiny (1092 people), and the queries are simply for topics (10 training topics and 50 test topics are provided).

6. CONCLUSION

We have prepared a large corpus that can act as a benchmark dataset for evaluating finding expert algorithms. We hope the availability of real-world data stimulates research on algorithms and systems, similar to what has been best practice in the Information Retrieval and Natural Language Technology fields with TREC, and in the area of Data Mining (e.g. UC Irvine datasets). We are interested in other sources (e.g. Digital Libraries) and possible expert finding scenarios to include in future releases of the ExpertFinder corpus.

7. REFERENCES

[1] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *WWW7: Proceedings of the seventh international conference on World Wide Web 7*, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[2] A. Gomez-Perez and A. Lopez-Cima. Portal ontology - knowledgeweb deliverable d1.6.2.

[3] C. Halaschek, B. Aleman-Meza, I. Arpinar, and A. Sheth. Discovering and ranking semantic associations over a large rdf metabase, 2004.

[4] A. Harth, J. Umbrich, and S. Decker. MultiCrawler: A Pipelined Architecture for Crawling and Indexing Semantic Web Data. In *5th International Semantic Web Conference*, 2006.

[5] A. Hogan, A. Harth, and S. Decker. ReConRank: A Scalable Ranking Method for Semantic Web Data with Context. In *2nd Workshop on Scalable Semantic Web Knowledge Base Systems*, 2006.

[6] H. Hwang, V. Hristidis, and Y. Papakonstantinou. Objectrank: a system for authority-based search on databases. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 796–798, New York, NY, USA, 2006. ACM Press.

[7] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *J. ACM*, 46(5):604–632, 1999.

[8] Y. Sure, S. Bloehdorn, P. Haase, J. Hartmann, and D. Oberle. The swrc ontology - semantic web for research communities. In C. Bento, A. Cardoso, and G. Dias, editors, *Proceedings of the 12th Portuguese Conference on Artificial Intelligence - Progress in Artificial Intelligence (EPIA 2005)*, volume 3803 of *LNCS*, pages 218 – 231, Covilha, Portugal, DEC 2005. Springer.

---

[3]http://sw.deri.org/svn/sw/2006/12/expertFinder/dbworld_jobs
[4]http://lists.w3.org/Archives/Public/www-rdf-interest/2004Dec/0015
[5]http://dublincore.org/
[6]http://knowledgeweb.semanticweb.org/semanticportal/OWL/
[7]http://www.aktors.org/publications/ontology/
[8]http://lsdis.cs.uga.edu/projects/semdis/swetodblp/

[9]http://trec.nist.gov/data/enterprise/05/ent05.expert.guidelines.final