Efficient reuse of structured and unstructured resources for ontology population

Chetana Gavankar, Ashish Kulkarni, Ganesh Ramakrishnan

Indian Institute of Technology - Bombay
chetanagavankar@gmail.com, kulashish@gmail.com, ganesh@cse.iitb.ac.in

Abstract

We study the problem of ontology population for a domain ontology and present solutions based on semi-automatic techniques. A domain ontology for an organization, often consists of classes whose instances are either specific to, or independent of the organization. *E.g.* in an academic domain ontology, classes like *Professor, Department* could be organization (university) specific, while *Conference, Programming languages* are organization independent. This distinction allows us to leverage data sources both — within the organization and those in the Internet — to extract entities and populate an ontology. We propose techniques that build on those for open domain IE. Together with user input, we show through comprehensive evaluation, how these semi-automatic techniques achieve high precision. We experimented with the academic domain and built an ontology comprising of over 220 classes. Intranet documents from five universities formed our organization specific corpora and we used open domain knowledge bases like Wikipedia, Linked Open Data, and web pages from the Internet as the organization independent data sources. The populated ontology that we built for one of the universities comprised of over 75,000 instances. We adhere to the semantic web standards and tools and make the resources available in the OWL format. These could be useful for applications such as information extraction, text annotation, and information retrieval.

Keywords: Ontology population, Semantic Web resources, Information Extraction

1. Introduction

An *ontology* describes entities in a domain and their interrelations. *Ontology population* concerns with the identification of instances and their mapping to classes and their attributes in an ontology. Such populated ontology is referred to as a knowledge base. Ontologies and knowledge bases play an important role in semantic web. This has led to an independent and distributed effort of developing several domain ontologies and public knowledge bases. An ontology for a domain can either be built from scratch or enriched using existing ontologies on the web. Search engines such as swoogle[1] allow to search for an existing ontology. We used such search engines to enrich existing academic ontologies by merging and extending them to incorporate classes from collaborative resources such as Wikipedia. We then populate the academic ontology using the intranet corpus, structured linked open data resources and the unstructured web data.

The paper is organized as follows. The following section discusses related work and compares them with our work. In section 2. we discuss the ontology building process using existing ontologies. The population of organization specific classes using list pages from an intranet corpus is explained in section 3.1. Further in section 3.2. and 3.3., we explain the process of ontology population of organization independent classes. We present the evaluation of our approach in section 4. followed by conclusion in section 5.

2. Ontology Building

Domain ontologies could either be built from scratch or extended from existing ontologies. We built our academic ontology using existing *Benchmark*[2] and *Aisso*[3] ontologies. Ontologies are merged using the *Protege*[4] ontology editor and extended to include several classes like *award*, *project* etc. and attributes like *professor* has *research-area*, *course* has *prerequisite* etc. In addition, we scraped the glossary lists available in wikipedia to populate class hierarchy rooted at the *concept* class. An ontology that we semi-automatically built, consists of more than 220 classes and 77,000 axioms. Please refer to figure 1 for a snapshot of the academic ontology. Currently we have populated the ontology with more than 75,000 instances from the linked open data resources, the web and a university corpus.

3. Ontology Population

We use the openly available resources including world wide web, linked open data along with intranet corpus to populate an academic ontology. Our semi-automatic approach extracts with high precision, entities to populate our academic ontology. Our ontology [5] is available as an open resource.

3.1. Ontology Population using list pages in an intranet corpus

In our academic ontology, we distinguish between two types of classes : in-domain and out-of-domain (also called domain independent) classes. In-domain classes are those whose instances can be populated from intranet corpus. Various information extraction techniques have been proposed that transform unstructured or semi-structured text to class-instance data. Here we follow a rule based approach

[1] http://swoogle.umbc.edu/

[2] http://swat.cse.lehigh.edu/onto/univ-bench.owl

[3] http://vocab.org/aiiso/schema

[4] http://protege.stanford.edu

[5] http://www.cse.iitb.ac.in/~chetana/AcadOnto.owl

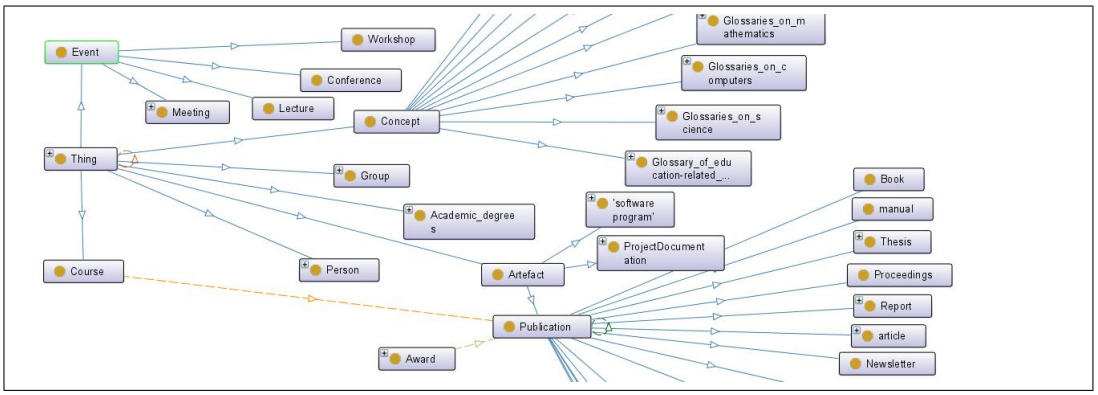

Figure 1: Academic Ontology snapshot of some classes in ontology

where we write annotators in a language called Annotator Query Language (AQL) [6]. Given an ontology, it is often not clear where and how to start writing annotators. This can be a tedious and complex task where the complexities arise from interdependencies amongst the concepts and ease (or the lack there of) of writing annotators for a concept before another. With the aim of understanding human judgment behind annotator writing and their ordering, we performed a manual exercise (Refer Appendix A) where we analyzed the rule writing process for higher level nodes and their leaf nodes. If a higher order concept has a very precise and obvious signature, then one would rather write that annotator first and perhaps use its output to help write lower-level annotators. An *address* annotator for instance might aid a *PIN number* annotator in precise extraction of PIN numbers. On the other hand, if such an obvious signature and/or rules are not present, then the composition approach of doing the properties and then combining them to high order concepts seems easier.

One of the key observations from this exercise was the need for glossaries. In bottom-up approach, the availability of glossary for each leaf concept in an ontology would help in writing accurate extractors for higher level ontology concepts. A *Professor information* annotator for instance will benefit from the availability of glossaries for *professor-name*, *department-name* and *course-name*. We describe the *Professor information* annotator implemented using AQL (Chiticariu et al., 2011) to illustrate our point. Assuming that the professor node can be populated using information on professor homepages, we first use a simple regular expression based extractor that looks for occurrence of the *homepage* word and filters professor homepages using the heuristic that the first name appearing on the homepage is that of its owner. We then use the glossary of professor names to extract professor name (Refer figure 2). We use the **span extraction operator** `extract` with dictionary construct for extracting spans of professor name on the page. AQL **relational operator** `select` was used along with `combine spans` construct to identify complete occurrence of professor name entity. The `union all` construct was then used to find combinations of names like *Gene Franklin*, *Gene F.*, *G.Franklin*, and *Franklin Gene*.

Professor's research area was extracted using the *research area* annotator. Here we use contextual phrases like *research area*, *research interest*, *area of interest* etc. to extract tokens occurring in proximity as the research area of the professor. A domain corpus is often replete with 'list pages'. In our academic corpus for instance, there are list pages containing list of departments, professors, courses, projects, research labs, events, and several others. Each of these correspond to an attribute of a leaf node in the academic ontology. The problem here is to locate these list documents for the ontology node of interest. Here we leverage the bootstrapping and learning to rank (Joachims, 2002) paradigms in an interactive setting. Posed as a learning to rank problem, the aim is to construct a ranking model that ranks list pages before others. Equipped with the ontology, we seek for search queries in the form of keywords and/or a seed list of instances for the node of interest. The tf-idf feature of these terms (using Lucene) in the corpus is used to obtain a partial ordering over the result set of documents. For each item in the set, we solicit a binary relevance judgment from users to indicate whether or not it is a list document. We extract instances from the identified list pages using rule based approach and visual DOM features. The evaluation of our approach is explained in the experiment section of the paper.

```
-- Rule R1a:Find dictionary matches for first names
create view ProfFirstName as
      extract dictionary 'ProfFirstNameDict'
      on D.text as fname from DetaggedDoc D;

-- Rule R1b: Similar to R1a to find dictionary matches for last names

--Rule R2: Initials of firstname, middlename or last names
create view ProfNameInitial as
      extract regex /([A-Z]\. ?)([A-Z]\. ?)?/
      on D.text as Nameinitial from DetaggedDoc D;

-- Rule R3: Combines output of R1a, R1b and R2 to extract full name
create view ProfName as
      -- Ex: Gene Franklin
      (select CombineSpans(F.fname, L.lname) as ProfessorName
      from ProfFirstName F, ProfessorLastName L
      where FollowsTok(F.fname, L.lname,0,0))
union all
      -- Ex: G.Franklin
      (select CombineSpans(F.Nameinitial, L.lname) as ProfessorName
      from ProfNameInitial F, ProfessorLastName L
      where FollowsTok(F.Nameinitial, L.lname,0,0))
union all
      --  Similar rules are written using above syntax for patterns
      --  like Franklin Gene, F. Gene, G.M.Franklin
```

Figure 2: Professor Name Annotator

[6] http://publib.boulder.ibm.com/infocenter/bigins/v1r3

3.2. Ontology Population using the web

The organization independent classes in an academic ontology are populated with instances from the web using implementation of SEAL (Set Expander for Any Language) (Wang and Cohen, 2007). SEAL is a set expansion system that takes as input a few seed instances of a target concept and then discovers other similar instances from semi-structured documents like web pages. In addition we also used structured linked open data resources to populate the organization independent in our ontology. The technique of using these collaborative resources is described in following section.

3.3. Ontology Population using linked open data resources

Linked Open Data (LOD) refers to interlinked, publicly available, and structured datasets on the web using semantic web standards. We search the required entities on linked open data to locate the relevant data source. Due to the openness of this LOD data sources, it is difficult to know data sources relevant for query answering. We use web interface, open link software[7] to ease the task of finding relevant data source. The results for a sample search for *glossary of mathematics* are displayed in the figure refer figure 3.

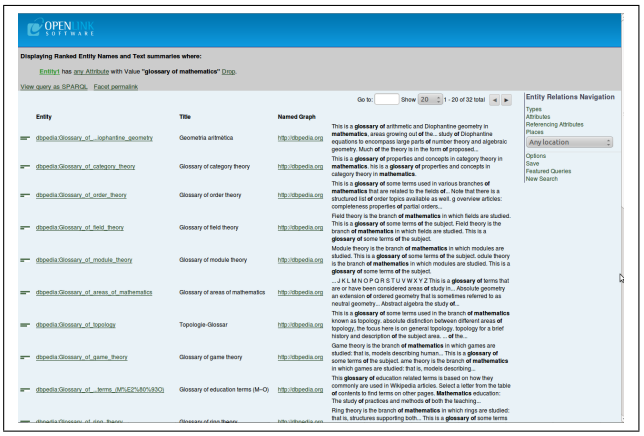

Figure 3: Link open data search results for glossary of mathematics

Subsequent to data source searching , we query these resources to extract the relevant instances. Data on the linked open data cloud are expressed using resource description framework (RDF) or web ontology language OWL. SPARQL protocol and RDF query language (SPARQL) can be used to express queries across diverse data sources. We populate our ontology by querying the linked open datasets using SPARQL for extracting the instances from these RDF resources on the LOD cloud. We wrote and executed SPARQL queries through DBpedia SPARQL endpoint [8]. Refer figure 4 for the results from a sample SPARQL query. The SPARQL queries return a set of instances to populate nodes in academic d open data. Resulting instances could then be used as seeds in spirit of the typical bootstrapping paradigm.

[7] http://dbpedia.org/fct/

[8] http://dbpedia.org/snorql/

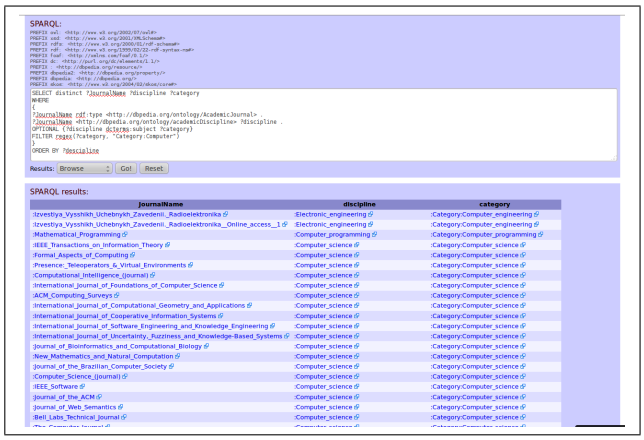

Figure 4: Sample SPARQL query execution

4. Experiments

The experiments were conducted for the academic ontology population using the three techniques described in the paper.

4.1. Evaluation using academic corpus

We generated our experimental corpus by crawling following university websites - Stanford University, Indian Institute of Technology Bombay (IITB) and Monash University - spanning different geographies. We evaluate our ranking model for identification of list pages. We also evaluate the two list extraction techniques - one that uses rule based approach and the other using visual features (DOM) for extraction of instances. The results for extraction using rule based annotator show that highly precise extraction can be achieved from list pages. Instances of classes that have a well defined signature like *email*, and *course-id* show close to 100% extraction accuracy. Others like *department-name*, *events*, and *research-lab* that exploit document features like page title, and contextual phrases, *etc.* also achieve high precision when run on list pages. The extractors for *person-name* and *research-area* additionally make use of negative word dictionaries comprising of common nouns, articles and conjunctions. The use of DOM path is motivated by the observation that all list items usually follow a similar DOM path within a document. We reused the potential list boxes as identified by cssbox and traced DOM paths within these that led to the seed list instances. We then extracted other instances by following the same path within these list boxes. The approach works very well achieving close to perfect precision and recall especially in vertically aligned lists. The results are summarized in table 4.1..

4.2. Evaluation using web resources

We implemented SEAL and used it to populate the classes in our academic ontology. We gave SEAL the benefit of knowing the list pages and then used it to extract instances from individual list page URLs. We report the results in table 4.2.[9]. While SEAL achieves 100% precision on most of the list pages, its recall is lower.

[9] SEAL failed to extract instances from some of the list pages

	Micro			Macro		
	P	R	F	P	R	F
IITB						
rule based	0.94	0.99	0.96	0.95	0.99	0.97
Visual features	0.98	0.89	0.93	0.97	0.96	0.96
Stanford						
rule based	0.88	0.98	0.94	0.77	0.99	0.87
Visual features	0.97	0.90	0.93	0.96	0.94	0.95
Monash						
rule based	0.96	0.99	0.97	0.93	0.99	0.96
Visual features	0.96	1	0.98	0.88	1	0.94

Table 1: list extraction using academic corpus

	Micro			Macro		
	P	R	F	P	R	F
SEAL	1	0.79	0.89	1	0.58	0.73

Table 2: Results of SEAL

4.3. Evaluation using linked open data resources

The purpose of evaluation was to ascertain the correctness of instances extracted from the linked open data for ontology population. We indexed corpus of three major universities obtained by crawling their pages. We then queried this index for each instance obtained from the linked open data and recorded the top 10 results. We scanned these results to check support for that instance in context of the category being populated. Figure 5 summarizes the results of our evaluation for a subset of classes in our academic ontology. We obtained precision close to one for most of the extracted instances from linked open data resources.

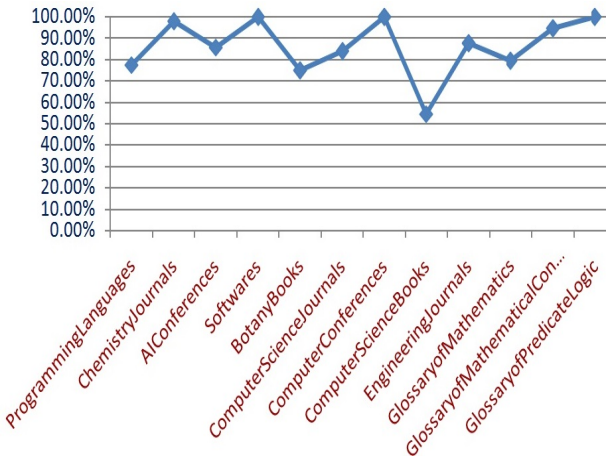

Figure 5: Precision calculated for some nodes populated using linked open data resources

5. Conclusion

Ontology captures domain knowledge in a particular area of interest, favoring interoperability and providing a shared understanding among the web-based applications (e.g. web services, resource sharing among enterprises, and in general, web information access). In semantic web ontology development and population are tasks of high importance. The manual performance of these tasks is labor- and therefore cost-intensive. In this paper we described the creation and population of academic ontology by effectively reusing the existing resources. We described the creation of ontology using existing ontologies on web, enriching with more classes from wikipedia and classes as observed from academic corpus. We presented various methods of semiautomatic population of academic ontology. We use existing collaborative resources on linked open data, instances from the web and the academic corpus list pages. Our results imply that the instances are relevant as observed through the high precision values for a set of instances for classes in our ontology.

6. References

Eugene Agichtein and Luis Gravano. 2000. Snowball: extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, DL '00, pages 85–94, New York, NY, USA. ACM.

Grigoris Antoniou and Frank van Harmelen. 2008a. *A Semantic Web Primer, 2nd Edition*. The MIT Press, 2 edition.

Grigoris Antoniou and Frank van Harmelen. 2008b. *A Semantic Web Primer, 2nd Edition (Cooperative Information Systems)*.

Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. 2010. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2nd edition.

Zhuowei Bao, Benny Kimelfeld, and Yunyao Li. 2012. Automatic suggestion of query-rewrite rules for enterprise search. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 591–600, New York, NY, USA. ACM.

Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The semantic web. *Scientific American*, 284(5):34–43.

A. Z. Broder and A. C. Ciccolo. 2004. Towards the next generation of enterprise search technology. *IBM Systems Journal*, 43(3):451 –454.

Marko Brunzel. 2008. The xtreem methods for ontology learning from web documents. In *Proceedings of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 3–26, Amsterdam, The Netherlands, The Netherlands. IOS Press.

R. Burget and I. Rudolfová. 2009. Web page element classification based on visual features. In *Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference on*, pages 67–72. IEEE.

Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. 2003. Extracting content structure for web pages based

on visual representation. In *Proceedings of the 5th Asia-Pacific web conference on Web technologies and applications*, APWeb'03, pages 406–417, Berlin, Heidelberg. Springer-Verlag.

Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors. 2012. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), Istanbul, Turkey, May 23-25, 2012*. European Language Resources Association (ELRA).

Soumen Chakrabarti, Sasidhar Kasturi, Bharath Balakrishnan, Ganesh Ramakrishnan, and Rohit Saraf. 2012. Compressed data structures for annotated web search. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 121–130, New York, NY, USA. ACM.

Tao Cheng, Xifeng Yan, and Kevin Chen-Chuan Chang. 2007. Entityrank: searching entities directly and holistically. In *Proceedings of the 33rd international conference on Very large data bases*, VLDB '07, pages 387–398. VLDB Endowment.

Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick R. Reiss, and Shivakumar Vaithyanathan. 2010a. Systemt: an algebraic approach to declarative information extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 128–137, Stroudsburg, PA, USA. Association for Computational Linguistics.

Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Frederick Reiss, and Shivakumar Vaithyanathan. 2010b. Domain adaptation of rule-based annotators for named-entity recognition tasks. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1002–1012, Stroudsburg, PA, USA. Association for Computational Linguistics.

Laura Chiticariu, Vivian Chu, Sajib Dasgupta, Thilo W. Goetz, Howard Ho, Rajasekar Krishnamurthy, Alexander Lang, Yunyao Li, Bin Liu, Sriram Raghavan, Frederick R. Reiss, Shivakumar Vaithyanathan, and Huaiyu Zhu. 2011. The systemt ide: an integrated development environment for information extraction rules. In *Proceedings of the 2011 international conference on Management of data*, SIGMOD '11, pages 1291–1294, New York, NY, USA. ACM.

Bhavana Bharat Dalvi, William W. Cohen, and Jamie Callan. 2012. Websets: extracting sets of entities from the web using unsupervised information extraction. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 243–252, New York, NY, USA. ACM.

Fernando M. B. M. de Castilho, Roger Granada, Breno Meneghetti, Leonardo Carvalho, and Renata Vieira. 2012. Corpus+wordnet thesaurus generation for ontology enriching. In Calzolari et al. (Calzolari et al., 2012), pages 3463–3467.

Pavel A. Dmitriev, Nadav Eiron, Marcus Fontoura, and Eugene Shekita. 2006. Using annotations in enterprise search. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 811–817, New York, NY, USA. ACM.

AnHai Doan, Raghu Ramakrishnan, and Shivakumar Vaithyanathan. 2006. Managing information extraction: state of the art and research directions. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, SIGMOD '06, pages 799–800, New York, NY, USA. ACM.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Commun. ACM*, 51(12):68–74, December.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ronald Fagin, Ravi Kumar, Kevin S. McCurley, Jasmine Novak, D. Sivakumar, John A. Tomlin, and David P. Williamson. 2003. Searching the workplace web. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 366–375, New York, NY, USA. ACM.

Wolfgang Gatterbauer and Paul Bohunsky. 2006. Table extraction using spatial reasoning on the css2 visual box model. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, AAAI'06, pages 1313–1318. AAAI Press.

Wolfgang Gatterbauer, Paul Bohunsky, Marcus Herzog, Bernhard Krüpl, and Bernhard Pollak. 2007. Towards domain-independent information extraction from web tables. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 71–80, New York, NY, USA. ACM.

David Hawking. 2004. Challenges in enterprise search. In *Proceedings of the 15th Australasian database conference - Volume 27*, ADC '04, pages 15–24, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

David Hawking. 2006. Enterprise search - the new frontier? In *ECIR'06*, pages –1–1.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 133–142, New York, NY, USA. ACM.

Gjergji Kasneci, Fabian M. Suchanek, Georgiana Ifrim, Shady Elbassuoni, Maya Ramanath, and Gerhard Weikum. 2008. Naga: harvesting, searching and ranking knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1285–1288, New York, NY, USA. ACM.

Yunyao Li, Rajasekar Krishnamurthy, Sriram Raghavan, Shivakumar Vaithyanathan, and H. V. Jagadish. 2008. Regular expression learning for information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 21–30, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiaonan Li, Chengkai Li, and Cong Yu. 2010. Entityengine: answering entity-relationship queries using shallow semantics. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1925–1926, New York, NY, USA. ACM.

Alain-Pierre Manine, Erick Alphonse, and Philippe Bessières. 2008. Information extraction as an ontology population task and its application to genic interactions. In *Proceedings of the 2008 20th IEEE International Conference on Tools with Artificial Intelligence - Volume 02*, ICTAI '08, pages 74–81, Washington, DC, USA. IEEE Computer Society.

Monica Marrero, Sonia Sanchez-Cuadrado, Jorge Morato, and Yorgos Andreadakis. 2009. Evaluation of Named Entity Extraction Systems. *Research In Computer Science*, 41:47–58.

Diana Maynard, Yaoyong Li, and Wim Peters. 2008. Nlp techniques for term extraction and ontology population. In *Proceedings of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 107–127, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Eyal Oren, Knud Mller, Simon Scerri, Siegfried Handschuh, and Michael Sintek. 2006. What are semantic annotations? Technical report, DERI Galway.

Massimo Poesio and Abdulrahman Almuhareb. 2008. Extracting concept descriptions from the web: the importance of attributes and values. In *Proceedings of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 29–44, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Ganesh Ramakrishnan, Soumen Chakrabarti, Deepa Paranjpe, and Pushpak Bhattacharya. 2004. Is question answering an acquired skill? In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 111–120, New York, NY, USA. ACM.

Lawrence Reeve and Hyoil Han. 2005. Survey of semantic annotation platforms. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 1634–1638, New York, NY, USA. ACM.

Giuseppe Rizzo and Raphael Troncy. 2011. Nerd a framework for evaluating named entity recognition tools in the web of data.

Warren Shen, AnHai Doan, Jeffrey F. Naughton, and Raghu Ramakrishnan. 2007. Declarative information extraction using datalog with embedded extraction predicates. In *Proceedings of the 33rd international conference on Very large data bases*, VLDB '07, pages 1033–1044. VLDB Endowment.

Stephen Soderland, Brendan Roof, Bo Qin, Shi Xu, Mausam, and Oren Etzioni. 2010. Adapting open information extraction to domain-specific relations. *AI Magazine*, 31(3):93–102.

Hyun-Je Song, Seong-Bae Park, and Se-Young Park. 2009. An automatic ontology population with a machine learning technique from semi-structured documents. In *Information and Automation, 2009. ICIA '09. International Conference on*, pages 534 –539, june.

Yuri A. Tijerino, David W. Embley, Deryle W. Lonsdale, Yihong Ding, and George Nagy. 2005. Towards ontology generation from tables. *World Wide Web*, 8(3):261–285, September.

Richard C. Wang and William W. Cohen. 2007. Language-independent set expansion of named entities using the web. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pages 342–350, Washington, DC, USA. IEEE Computer Society.

Tim Weninger, Fabio Fumarola, Rick Barber, Jiawei Han, and Donato Malerba. 2011. Unexpected results in automatic list extraction on the web. *SIGKDD Explor. Newsl.*, 12(2):26–30, March.

Huaiyu Zhu, Sriram Raghavan, Shivakumar Vaithyanathan, and Alexander Löser. 2007. Navigating the intranet with high precision. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 491–500, New York, NY, USA. ACM.

A Annotator Writing approach

Given a domain ontology, a knowledge engineer will benefit from the knowledge of whether to write an annotator (for a concept) top-down or bottom-up. In the bottom-up approach, annotators for lower level concepts are written first and then aggregated (using a higher level operation) to write an annotator for a higher level domain concept. In the top-down approach on the other hand, an annotator for an intermediate concept is written first. That knowledge is leveraged in coming up with lower level annotators.

Typically the decision on the order of writing annotators is taken by a human. In an attempt to check if this decision can be automated, we performed an independent exercise where we documented the human judgment that underlies the annotator writing process. What follows is a sample of concepts from academic (and technical) domain and a short write-up on the approach for writing an annotator for them.

Course ID

A course ID seems to follow a fixed pattern. A study of various universities showed that the pattern is specific to that university but follows a predefined rule. Stanford for instance has an alphanumeric course ID where the first two letters indicate the department followed by three digits that identify if it is an undergraduate/graduate course and the specific area. *e.g.* CS101. It is also observed that the ID is between 5-7 characters long.

It is possible to write a regex based extractor for course ID that builds on the combination of above knowledge. Alternatively, dynamically built dictionaries can also be exploited. For instance, a dictionary of department Ids would be useful in extracting course Ids for Stanford or IITB. Each department in the university also maintains a directory listing of all the offered courses. So yet another way could to be build a dictionary of course Ids. Such a dictionary could be exploited in writing extractors for higher level concepts like *Course*. A bottom-up approach seems natural in this case.

Research Project

A research project consists of members, supervisor, domain of work, a research topic, set of artifacts etc. Each department of an university seems to maintain a listing of its research projects and this is consistent across universities. The page typically lists the project name along with a short description and a link for further details. The project name can be any free text and is not observed to follow a pattern. It is non trivial to extract a project name without the knowledge that the given page concerns research projects. This knowledge can be acquired by looking at words like *Project, Research, and Resource* in the page URL, breadcrumbs, or the title.

This calls for a top-down approach where a *ResearchProject* concept annotator should be written before writing annotators for its constituent nodes. It is observed that the project page is not always complete especially in the listing of its contributors. The project membership and perhaps other relationships could be spread across the university web site. A dynamic dictionary of the project names could therefore be useful. Thus the typical ordering of annotators in this case could be *ResearchProject* followed by *'project name', 'supervisor', 'members'* etc where the project name dictionary is exploited in annotating concepts like project members.