

CONFERENCES IN RESEARCH AND PRACTICE IN
INFORMATION TECHNOLOGY

VOLUME 85

ADVANCES IN ONTOLOGIES 2007



AUSTRALIAN
COMPUTER
SOCIETY

ADVANCES IN ONTOLOGIES 2007

Proceedings of the
3rd Australasian Ontology Workshop (AOW 2007),
Gold Coast, Australia, 2 December 2007

Thomas Meyer and Abhaya C. Nayak, Eds.

Volume 85 in the Conferences in Research and Practice in Information Technology Series.
Published by the Australian Computer Society Inc.



Published in association with the ACM Digital Library.

Advances in Ontologies 2007. Proceedings of the 3rd Australasian Ontology Workshop (AOW 2007), Gold Coast, Australia, 2 December 2007

Conferences in Research and Practice in Information Technology, Volume 85.

Copyright © 2007, Australian Computer Society. Reproduction for academic, not-for-profit purposes permitted provided the copyright text at the foot of the first page of each paper is included.

Editors:

Thomas Meyer
Meraka Institute
PO Box 395
Pretoria 001
South Africa
E-mail: tommie.meyer@meraka.org.za

Abhaya C. Nayak
Intelligent Systems Group (ISG)
Department of Computing
Macquarie University
Sydney, NSW 2109
Australia
E-mail: abhaya@comp.mq.edu.au

Series Editors:

Vladimir Estivill-Castro, Griffith University, Queensland
John F. Roddick, Flinders University, South Australia
Simeon Simoff, University of Technology, Sydney, NSW
crpit@infoeng.flinders.edu.au

Publisher: Australian Computer Society Inc.
PO Box Q534, QVB Post Office
Sydney 1230
New South Wales
Australia.

Conferences in Research and Practice in Information Technology, Volume 85
ISSN 1445-1336
ISBN 978-1-920682-66-8

Printed March 2008 by Flinders Press, PO Box 2100, Bedford Park, SA 5042, South Australia.
Cover Design by Modern Planet Design, (08) 8340 1361.

The *Conferences in Research and Practice in Information Technology* series aims to disseminate the results of peer-reviewed research in all areas of Information Technology. Further details can be found at <http://crpit.com/>.

Table of Contents

Proceedings of the 3rd Australasian Ontology Workshop (AOW 2007), Gold Coast, Australia, 2 December 2007

Preface	vii
Programme Committee	viii
Acknowledgement of Support	ix

Keynote Paper

The Development, Evaluation and Application of Ontologies to eResearch	3
<i>Jane Hunter</i>	

Full Papers

Enterprise Semantic Information Search System Based on New Music and Audio Ontology Integrating Existing Ontologies	7
<i>Kiavash Bahreini, and Atilla Elçi</i>	
Learning from Ontological Annotation: an Application of Formal Concept Analysis to Feature Construction in the Gene Ontology	15
<i>Elma Akand, Michael Bain, and Mark Temple</i>	
Structure Based Semantic Measurement for Information Filtering Agents	25
<i>Glenn Boardman, and Hongen Lu</i>	
An ontology-based approach for resolving semantic schema conflicts in the extraction and integration of query-based information from heterogeneous web data sources	35
<i>Abdolreza Hajmoosaei, and Sameem Abdul Kareem</i>	
A Formalization of Subjective and Objective Time Ontologies	45
<i>Philip H.P. Nguyen, and Dan Corbett</i>	
Dealing with the Formal Analysis of Information Security Policies through Ontologies: A Case Study	55
<i>Geiza M.H. da Silva, Alexandre Rademaker, Davi Romero de Vasconcelos, Fernando N. Amaral, Carlos Bazilio, Vaston Costa, and Edward Hermann Haeusler</i>	
Extraction, evaluation and integration of lexical-semantic relations for the automated construction of a lexical ontology	61
<i>Tonio Wandmacher, Ekaterina Ovchinnikova, Ulf Krumnack, and Henrik Dittmann</i>	
Author Index	71

Preface

The first two Australasian Ontology Workshops (AOW 2005 and AOW 2006) were held in Sydney and Hobart, respectively, both as workshops of the Australian Joint Conference on Artificial Intelligence (AI'05, and AI'06). This tradition is being continued this year, with AOW 2007 being held on the Gold Coast in Queensland, Australia, again as workshop of the Australian Joint Conference on Artificial Intelligence (AI'07).

The purpose of this one-day workshop series on Advances in Ontologies is to bring together ontology researchers from both industry and academia in the Australasian region for interaction, discussion, sharing of results and initiation of new projects, and also to raise the awareness of the Australasian Artificial Intelligence community to the state-of-the-art ontology research conducted in the region. AOW 2007 has in particular provided a visible focal point for ontology research within the Australasian region, and provided a connection with the international ontology community.

The keynote speaker, Professor Jane Hunter from the University of Queensland, elaborated her hypothesis that the application of semantic web technologies to the semantic annotation, integration and correlation of distributed mixed-media scientific datasets and data processing services can expedite the discovery of new knowledge - that scientific problems can be solved more quickly through richer, machine-processable descriptions, enhanced semantic interoperability and faster data integration. She presented three interesting e-Research applications in support of this claim.

A program committee of international standing reviewed all contributed papers (full papers were reviewed). Each paper was reviewed by at least three program committee members, and additional reviews were also sought to identify those papers which propose the most promising ideas. As a result, seven papers were selected for publication in these proceedings out of eleven submitted papers by authors from Australia, Brazil, Germany, Malaysia, Turkey and the United States.

The papers in this issue deal with varied aspects of ontology research, including semantic annotation, semantic measures, lexical ontology and temporal ontology. They discuss issues involved in the construction of ontologies such as integration of relations, and the resolution of schema conflicts. As well, there are discussions of the challenges arising out of the applications of ontological research to different areas, including formal analysis of information security policies.

We would like to thank the keynote speaker, Jane Hunter, the authors and the members of the Program Committee of AOW 2007 and the additional reviewers for their contributions to the quality of the workshop and of this collection.

Thanks are also due to the members of the AI'07 Organising Committee — in particular Dr Marcus Randall — for their help with the smooth organisation of this workshop event, and the editors of the CRPIT series for facilitating the publication of the AOW 2007 workshop proceedings. We acknowledge the EasyChair conference management system which was used in all stages of the paper submission and review process and also in the collection of the final camera-ready papers.

Thomas Meyer, Meraka Institute
Abhaya C. Nayak, Macquarie University
Organisers of AOW 2007
December, 2007

Programme Committee

Programme Chairs

Thomas Meyer (Meraka Institute, South Africa)
Abhaya C. Nayak (Macquarie University, Australia)

Programme Committee

Mike Bain (UNSW, Australia)
Richard Booth (Mahasarakham University, Thailand)
Werner Ceusters (SUNY Buffalo, USA)
Anne Cregan (UNSW, Australia)
Atilla Elçi (Eastern Mediterranean University, Turkey)
Joerg Evermann (Victoria University Wellington, New Zealand)
Aurona Gerber (CSIR, South Africa)
Manolis Gergatsoulis (Ionian University, Greece)
Dennis Hooijmaijers (University of South Australia, Australia)
Bo Hu (University of Southampton, UK)
Renato Iannella (NICTA, Australia)
Laurent Lefort (CSIRO, Australia)
Costas Mantratzis (University of Westminster, UK)
Lars Mönch (University of Hagen, Germany)
Deshendran Moodley (University of KwaZulu Natal, South Africa)
Mehmet Orgun (Macquarie University, Australia)
Bhavna Orgun (Macquarie University, Australia)
Maurice Pagnucco (UNSW, Australia)
Anet Potgieter (University of Cape Town, South Africa)
Debbie Richards (Macquarie University, Australia)
Rolf Schwitter (Macquarie University, Australia)
Rajan Shankaran (Macquarie University, Australia)
Barry Smith (SUNY Buffalo, USA)
Markus Stumptner (University of South Australia)
York Sure (SAP Research, Germany)
Kerry Taylor (CSIRO, Australia)
Mary-Anne Williams (UTS, Australia)

Additional Reviewers

Jennifer Fang
Yuan-Fang Li
Christos Papatheodorou

Acknowledgement of Support

We wish to thank the Meraka Institute and Macquarie University for their continuing support that made the organisation of this workshop possible.

We also acknowledge Griffith University for the local arrangements, as well as the local organization team of the Twentieth Australian Joint Conference on Artificial Intelligence 2007.



Meraka Institute
CSIR Site - Building 43
Meiring Naude Road
Brummeria
Pretoria
South Africa
<http://www.meraka.org.za>



MACQUARIE
UNIVERSITY ~ SYDNEY
Macquarie University
Sydney, NSW 2109, Australia
<http://www.mq.edu.au/>



Griffith University
QLD 4111, Australia
<http://www.griffith.edu.au/>

KEYNOTE PAPER

The Development, Evaluation and Application of Ontologies for eResearch

Jane Hunter

School of ITEE
The University of Queensland
St Lucia, Queensland

j.hunter@uq.edu.au

Extended Abstract

Advances in scientific research techniques have led to an explosion of information-rich, multimedia data within the research sector. New high-throughput data capture and combinatorial experimentation techniques (involving advanced instruments capable of capturing extremely high resolution data streams) have resulted in the generation of research data in quantities that are too great for effective assimilation. The data is not only massive in volume but is also being produced in a broad range of mediums and formats, including: numerical data, spectrographic output, genomic arrays, images, 3D models, audio and video, for disciplines including nano-materials, bioinformatics, tele-medicine, geosciences, astronomy and the social sciences. Scientific discovery is increasingly dependent on reliable tools and services to support the storage, dissemination, analysis and correlation of these complex data sets by collaborating teams of globally distributed scientists.

The volume, variety and multi-dimensional nature of the content exacerbates the difficulty of describing this data adequately so it can be confidently and appropriately incorporated into existing theories or models. In order to validate and authenticate scientific results, detailed provenance metadata describing the precise methodology and derived datasets needs to be recorded. Because today's scientists are working in large geographically distributed teams or "virtual organisations", the data and metadata has to be comprehensible to people, computers and software across many different organizations, platforms and disciplines. Metadata standards and semantic interoperability are essential to enable distributed querying, analysis, integration of mixed-media, heterogeneous scientific datasets in order to maximize its re-use, extract the inherent knowledge and build new knowledge layers on top of existing data

The Semantic Web promotes interoperability through formal languages and rich semantics. It aims to build a web where information is exchanged easily between humans and machines. Through a combination of URIs, RDF, OWL ontologies, SWRL inferencing rules and SPARQL query language, the Semantic Web aims to

define and expose the semantics associated with data or information, in order to facilitate automatic processing, integration, sharing and reuse of the data.

The hypothesis we are trying to prove is that the application of semantic web technologies to the semantic annotation, integration and correlation of distributed mixed-media scientific datasets and data processing services, offers enormous potential for expediting the discovery of new knowledge. Semantic web/grid tools enhance interoperability through formal syntaxes, ontologies and inferencing rules. They enable innovative search, data exploration, hypothesis development and evaluation interfaces and can assist researchers in managing, assimilating and distributing data to facilitate further scientific understanding and discovery.

In this paper we present three e-Research applications that support this hypothesis – they demonstrate three disciplines in which scientific problems may be solved more quickly through: richer, machine-processable descriptions, enhanced semantic interoperability and faster data integration:

- Fuel Cell Optimization (Hunter 2004)
- Semantic WildNet (Pullar 2007)
- Ethnographic Media Analysis (Schroeter 2006)

In particular our approach is to facilitate semantic interoperability across media types, vocabularies and disciplines through a common extensible ontology (Hunter 2003). The significant advantage of this approach is that it can easily be extended and adapted across disciplines through the incremental incorporation of domain-specific ontologies and rules.

References

- Hunter, J. Drennan, J. Little, S. (2004) "Realizing the Hydrogen Economy through Semantic Web Technologies", IEEE Intelligent Systems - Special Issue on eScience, Jan-Feb 2004
- Pullar, D., Zhang, J. Hunter, J. Zhou X. (2007) *Integrative Environmental Queries Using Geospatial Web Services*, Distributed Geoinformatics and Sensing, Ubiquity, and Mobility (DG/SUM07), Sept 2007
- Schroeter, R., Hunter, J., et. Al. (2006) *A Synchronous Multimedia Annotation System for Secure Collaboratories*, 2nd IEEE International Conference on E-Science and Grid Computing, Dec 2006
- Hunter, J. (2003). *Enhancing the Semantic Interoperability of Multimedia through a Core Ontology*. IEEE Trans. on CSVT, Feb 2003

FULL PAPERS

Enterprise Semantic Information Search System Based on New Music and Audio Ontology Integrating Existing Ontologies

Kiavash Bahreini and Atilla Elci

Dept. of Computer Engineering, and Internet Technologies Research Centre,
Eastern Mediterranean University, Famagusta, TRNC, Turkey

Email: {kiavash.bahreini, atilla.elci}@emu.edu.tr

Abstract

Music and Audio Information Search System (MAISS) is a web-based application using ontologies and inference engine to search multimedia documents. In MAISS, users run queries in web pages for retrieving data about albums, artists, audio files, audio file formats, encoding audio files, genre, instrument, key, note, official, resource, rhythm, etc. Web Ontology Language (OWL) is the operational base of MAISS, so users can also run queries for retrieving information with constraints about classes, data type properties, object properties, and their values. MAISS shows many categories of information about music and audio files. In fact this system is a database of information about music which enables the user to obtain information roughly or accurately. This system is not only a machine-readable system and capable of converting information from OWL format to RDF format but also it can extract data from ontology file whereby it would be user-readable and understandable. Moreover, for implementing this system, java language, J2EE architecture, and other related technologies in addition to Semantic Web have been used.

Keywords: Ontology, OWL, RDF, RDQL, J2EE, Music and Audio.

1 Introduction

The music industry is changed by Internet as it rendered music easy to share, listen to, sell and buy. Information about music, musicians, their relationship or experience can be shared on the Internet. Tracks may be in one or multiple formats. Holding and serving such data is not possible unless systems or sites exist for supporting them.

Music is an art form that involves organized and audible sounds and silence (MW 2007). Music may be used for artistic or aesthetic, communicative, entertainment, or ceremonial purposes. This project is chiefly focusing on the music information. The MAISS is a system which helps people to reach to information about music.

Users can search and find about Artists, Albums,

Instruments, Tracks, Rhythms, Keys, Audio file types, Audio file formats, etc. The MAISS is based on Web Ontology Language (OWL 2004) and is capable of extracting data from ontology file.

Our goal in this project is to establish a new ontology database using OWL on which rule-based inference engines like Jena (Jena 2006) are able to run queries and return requested data. It acts like a small search engine which returns related information and links about users' requests based on music system.

The Music Ontology Specification provides main concepts and properties for describing music in OWL on the Semantic Web (MOS 2006). The emerging idea of the Semantic Web is based on the maximum automation of the complete knowledge lifecycle processes, i.e., knowledge representation, acquisition, adaptation, reasoning, sharing, and use (Stamou 2005). In Semantic Web, the machines must be able to discover common meanings. A solution to this problem is introduced by "Ontologies" (OASF: Gayde et al. 2006). OWL uses both URIs for naming and the description framework for the Web resources provided by RDF to add the following capabilities to ontologies (OWL 2004):

1. Ability to be distributed across many systems.
2. Scalability to Web needs.
3. Compatibility with Web standards for accessibility and internationalization.
4. Openness and extensibility.

Although OWL has such capabilities these are not enough for its use in web-based applications or conceptual systems. Furthermore, many related web sites which are constructed for music ontology are not in OWL yet; most are dealing with music ontology in only RDF format. They survey music art and industry but just literature review on these topics are not enough. Moreover, with the advancement of technology, the changing needs for getting better results may be served by using new technologies and mixing them with many developing techniques in computer science. There are many APIs which can be used to obtain better inference capability. Applying them on ontology can improve our ability for better working, running powerful queries, getting better results, and so on.

In order to be applicable for real-world enterprise applications, our ontology representation approach must make it easy to fulfil the following technical requirements:

Copyright © 2007, Australian Computer Society, Inc. This paper appeared at *3rd Australasian Ontology Workshop (AOW-07)*, Gold Coast, Queensland, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 85. Editors, Thomas Meyer and Abhaya C. Nayak. Reproduction for academic, not-for profit purposes permitted provided this text is included.

1. Scalability: systems must be able to cope with large quantities of information.
2. Concurrency support: it must be possible for several users to use and read information at the same time.
3. Reliability: the system must under no circumstances lose or corrupt information.
4. Easy integration with existing data sources.

These requirements are not trivial to fulfil, largely due to the fact that ontology management infrastructure has not reached the maturity of the relational databases. For example, many existing tools are still file-oriented and so is ours. This limits the size of ontologies that can be processed, as the whole ontology must be read into main memory. As our project is web based, it uses multi-tier architecture, so it is able to support multi-user transactions. It uses RDQL (Jena Tutorial 2004) for running and returning results.

Although semantics has been used and placed in OWL files and it has given some virtual meaning to those data but we still need more efficient information system so that the data would be accessible for both human and machine. As a matter of fact we need to implement some systems so that users all over the world would be able to retrieve data from ontology-based database(s) via Internet and this system would enable the user to save retrieved information with their selected format.

Semantic information systems often have the following problems to deal with:

- Implementing multiuser transaction queries over files of ontology simultaneously,
- Understanding the existing data within the ontology by the user,
- Lack of existing web-based applications to work with Semantic Web, and
- Difficulty of converting the Semantic Web formatting languages through the web interface.

All above mentioned problems have been addressed by MAISS system.

The remainder of the paper is organized as follows:

Section 2 displays an overview of MAISS. Section 3 investigates architecture of MAISS which is based on ontology definition in OWL, inference engine, and J2EE n-tier model. In section 4, classes in music ontology and ontology specification are explained. Section 5 considers many queries which were used in Algernon or through JSP web pages. Section 6 concludes the paper and suggests future work in this topic.

2 MAISS Overview

MAISS maintains an ontology for representing knowledge on multimedia besides holding multimedia information and data (MAISS: Bahreini et al. 2007).

MAISS has a graphical user interface which, through web pages, represents, and saves data and knowledge. It displays records related to OWL database and is able to save data in RDF (RDF 2004) format. The MAISS, for gaining this functionality, uses computation and inference engines.

How does the MAISS work? As soon as a request is received from the user it is passed on to the computation engine [CE in abbreviation] (reference step no. 1 in Figure 1); CE generates proper queries for posing them to the inference engine [IE in abbreviation] (step no. 2). IE checks the Ontology & Knowledge Base, receiving related data (steps no. 3 & 4). IE extracts and forwards relevant information to CE. Finally, CE renders the information for displaying through the user interface.

Operational steps of MAISS query processing are displayed in Figure 1.

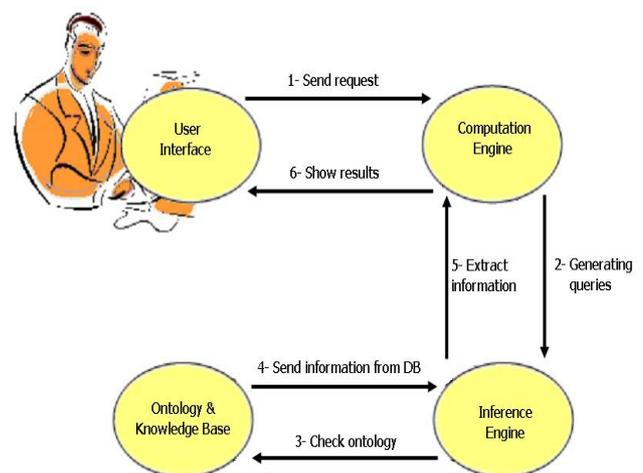


Figure 1: MAISS query processing.

3 MAISS Architecture

MAISS is an ontology-based linguistic music and audio search engine, developed using OWL, an inference engine, J2EE (J2EE 1.4), and Web. Currently, MAISS source code contains more than 6,000 lines of OWL, JSP, and Java code.

The base of MAISS architecture is J2EE n-tier model. Java Platform, Enterprise Edition (Java EE) is the industry standard for developing portable, robust, scalable and secure server-side Java applications. Building on the solid foundation of Java SE, Java EE provides web services, component model, management, and communications APIs that make it the industry standard for implementing enterprise applications (Algernon 2005). This architecture is shown in Figure 2. In this model there are many tiers; Client Tier is in the first part. All browsers can be used in this tier. User's request is conveyed to the next tier, Presentation Tier. The web server runs in this tier providing JSP and Servlet containers. Presentation Tier is able to receive requests from client tier and call query methods to execute at the next tier, Business Logic Tier. Java Server Pages (JSP) technology provides a simplified and fast way to create dynamic web content. JSP technology enables rapid

development of web-based applications that are server- and platform-independent (JSP 2006). Java Servlet technology provides Web developers with a simple, consistent mechanism for extending the functionality of a Web server and for accessing existing business systems. A servlet can be thought of as an applet that runs on the server side--without a face. Java servlets make many Web applications possible (Servlet 2006).

Business Logic Tier is at the heart of query processing which make up the MAISS. All of the JavaBeans, business components and Jena API exist here. JavaBeans technology is the component architecture for the Java 2 Platform, Standard Edition (J2SE). Components (JavaBeans) are reusable software programs that can be developed independently and assembled easily to create sophisticated applications. JavaBeans technology is based on the JavaBeans specification (JavaBeans 2006). Jena is a Java framework for building inference capability into Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL, and includes a rule-based inference engine. It is able to generate queries for the Integration Tier.

The RDQL exist in the Integration Tier. It runs queries on the Data Tier and extracts result sets to the Business Logic Tier. RDQL is a query language for RDF in Jena models. The idea is to provide a data-oriented query model so that there is a more declarative approach to complement the fine-grained, procedural Jena API (Jena Tutorial 2004).

The Data Tier contains the MAISS Music Ontology file also storing our data and knowledge.

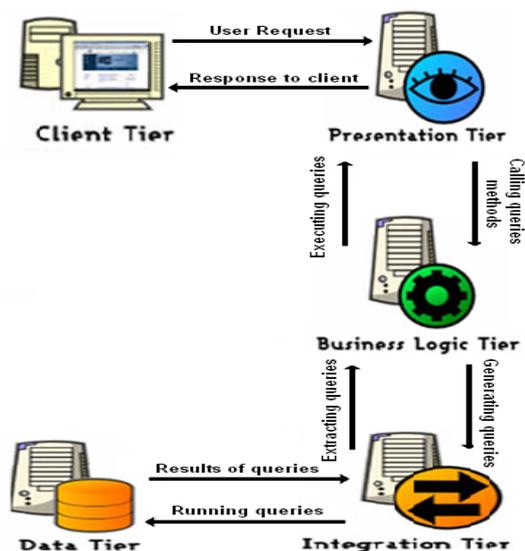


Figure 2: MAISS Architecture; relations between tiers.

The MAISS Music Ontology is introduced in detail in the next section.

4 MAISS Music Ontology

An ontology in Semantic web is used to model the vocabulary and meaning of the domains. That is to say, the objects, the relationships between them, the

properties, functions, constraints and rules are defined and modelled by an ontology. It includes machine-interpretable definitions of basic concepts in the domain and relations among them (OASF: Gayde et al. 2006). The wide usage of Semantic Web and the number of people contributing to the web increase numbers of ontologies. One of the basic problems in the development of the Semantic Web is the integration of ontologies (IOMSP: Olgu et al. 2006).

MAISS is an ontology-based linguistic music and audio search engine, developed using OWL, inference engine, J2EE, and Web. We have developed a Music and Audio Ontology to provide a semantic framework for MAISS. It is expressed in Web Ontology Language (OWL 2004). MAISS Music Ontology is built starting with the Music Ontology Specification (MOS 2006) and also considerably extended it; it currently has more than 30 classes and 100 objects and data type properties.

In this paper we describe many classes which are available in the Music Ontology Specification in its initial version (MOS 2006):

1. Artist: this is a generic term which is applied to solo artists, groups, and also "various artists".
2. AudioFile: an archived digital signal.
3. AudioFileType: the archiving type used.
4. Encoding: the encoding used in the archiving process.
5. EP: an EP is a so-called "Extended Play" release and often contains the letters EP in the title.
6. Form: anchor point for musical form taxonomy.
7. Key: Musical keys.
8. Longplay: a "Long Play" (LP) release (Album), generally consists of previously unreleased material. This includes release re-issues, with or without bonus tracks.
9. Note: a musical note.
10. Opus: the abstraction of a musical piece.
11. Other: any release that does not fit or cannot decisively be placed under any of the other categories.
12. Resource: web resources.
13. Rhythm: rhythm of music.
14. Score: a transcription of a musical piece (may be produced by an arrangement).
15. Signal: a digital or an analog signal.
16. Single: a single typically has one main song and possibly a handful of additional tracks or remixes of the main track. A single is usually named after its main song.
17. Status: album release (Album) status. This is the super class of all classes' status.
18. Type: a type of Album release (Album). This is the super class of all classes' types.

The sample diagram displaying ontology classes of MAISS, as generated by TGViz API in Protégé, is shown in Figure 3. The name space prefix “mo:” refers to the Musical Ontology Specification (MOS 2006); the rest are indigenous to MAISS.

There are many classes and properties in our OWL file which is very big to list here, but as an example, the Artist class and the image property which has Artist as one of its domain are depicted in Table 1.

```
<owl:Class rdf:ID="Artist">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#MusicAudio"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int">
1</owl:maxCardinality>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="age"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:comment
rdf:datatype="http://www.w3.org/2001/XMLSchema#string
">Artist - This is a generic term which is applied to solo
artists, groups, and also "various artists".
</rdfs:comment>
</owl:Class>
<hr/>
<owl:ObjectProperty rdf:ID="image">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Artist"/>
        <owl:Class rdf:about="#Album"/>
        <owl:Class rdf:about="#Track"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="#Resource"/>
  <rdfs:comment
rdf:datatype="http://www.w3.org/2001/XMLSchema#string
">image - Indicates a pictorial image (JPEG, GIF, PNG,
Etc.) of an artist, an album or a track.</rdfs:comment>
</owl:ObjectProperty>
```

Table 1: The Artist class and image property declaration in OWL file.

In the following section some queries and their resultant output will be shown.

5 Queries in MAISS

Search engines are very important tools for the people to get information from Internet but low-accuracy and low-recall persist widely in current search engines (DSSWS: Çelik et al. 2006). Certainly query facilities are of critical importance for any ontology-based information system. It is important that results of queries reflect the original semantics of the model. The results of queries we attempted are related to the original semantic and are also user readable.

For applying the Artist concept, we use subjects, verbs,

and objects for retrieving data from MAISS Music and Audio Ontology. For example for retrieving web link for images of artists, we implemented the query in triples (the subjects, verbs, and objects that make up RDF statements) using RDQL in Java language which is shown in Table 2.

```
...
loaded_model.read(new InputStreamReader(in, ""));
String queryString = " SELECT ?" + Artist +
" WHERE ( ?" + Artist + ", nss:" + image + ", ?" +
objectName + ")" +
" USING nss FOR <" + NS + ">";
Query query1 = new Query(queryString);
query1.setSource(loaded_model);
QueryExecution qe = new QueryEngine(query1);
QueryResults results = qe.exec();
...
```

Table 2: The query in RDQL; resource for image of Artists.

The result of the above query in browser window is shown in Figure 4.

In the following we show an example using Algernon (Algernon 2005); this query returns information about each Artist having numberOfAlbums greater than 25 (See Table 3):

```
((:INSTANCE Artist ?aName)
(age ?aName ?age)
(nationality ?aName ?nationality)
(country ?aName ?country)
(numberOfAlbums ?aName ?numberOfAlbums)
(:FAIL (:neq ?age ?age))
(:FAIL (:neq ?nationality ?nationality))
(:FAIL (:neq ?country ?country))
(:FAIL (:neq ?numberOfAlbums ?numberOfAlbums))
(:FAIL (:TEST (:LISP (> 25 ?numberOfAlbums))))))
```

Table 3: The query in Algernon; information about each artist having numberOfAlbums greater than 25.

The result of the above query in Algernon is shown in Figure 5.

Table 4 shows the query which is about artists who’s ‘age + albums’ is less than 50:

```
((:INSTANCE Artist ?name)
(age ?name ?age)
(numberOfAlbums ?name ?Albums)
(:BIND ?AgeAlbums (:LISP (+ ?age ?Albums)))
(:FAIL (:TEST (:LISP (< 50 ?AgeAlbums))))))
```

Table 4: The query in Algernon; artist’s age + album is less than 50.

Next query, which is shown in Table 5, displays file type of AudioFiles. They can be in mp3, wma, vox etc format. For executing this query we call one method called runMyQuery (...) with three input arguments; these parameters are used in queryString object:

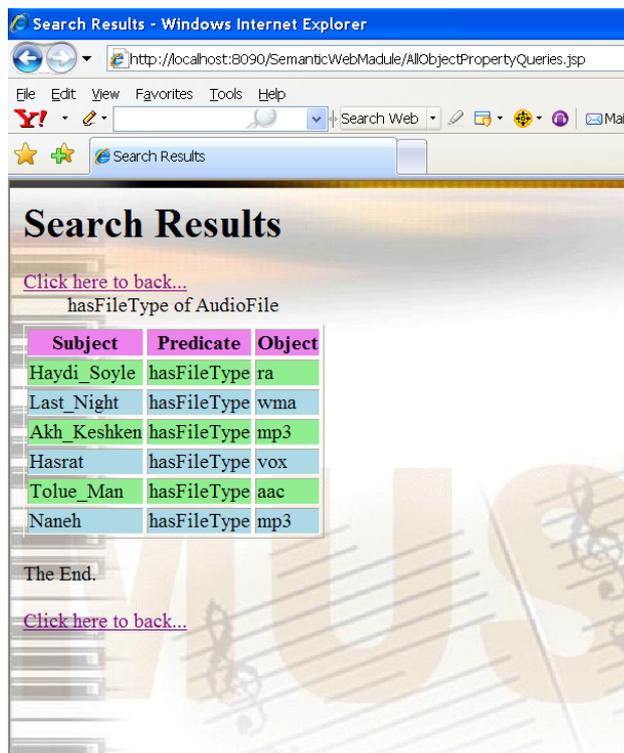
```

runMyQuery(ricievedSubjaectName, "hasFileType", "a");
...
public void runMyQuery(String subjectName, String
predicateName, String objectName) {
...
String queryString = " SELECT ?" + subjectName
+"WHERE(?"+subjectName+"nss:" + predicateName + ",
?" + objectName + ")" +" USING nss FOR <" + NS + ">";
Query query1 = new Query(queryString);
query1.setSource(loaded_model);
QueryExecution qe = new QueryEngine(query1);
QueryResults results = qe.exec();
...
}

```

Table 5: The query in RDQL; file type of AudioFiles.

The result of above query in browser window is depicted in Figure 6.



Subject	Predicate	Object
Haydi_Soyle	hasFileType	ra
Last_Night	hasFileType	wma
Akh_Keshken	hasFileType	mp3
Hasrat	hasFileType	vox
Tolue_Man	hasFileType	aac
Naneh	hasFileType	mp3

Figure 6: The snapshot of the triple; file type of AudioFiles.

One of the basic problems of the Semantic Web is integration of ontologies. Indeed, the web includes variety of information however in order to extract and combine information, say in a summary document, semantic integration is required. If the ontologies of the web pages can be integrated into a virtual ontology (VO), then only that would be searched (IOMSP: Olgu et al. 2006). Virtual ontology concept is applied in this research.

In a regular database, there is no class relationship. Therefore sub classes cannot inherit all properties from their super classes.

Extra work will be required to define class relations and all properties separately. Thus, the work done in SW is less. The class hierarchies and properties are well defined. Therefore, doing search with rule-based inference engine eases the work done and yields better performance. Each class in a relational database model is represented as a separate entity. In Semantic Web, information is represented by using triples (subject, predicate, object) whereas in relational database a record is an RDF or OWL node, the column name is RDF or OWL propertyType, and the record field is a value. Doing inference through relational method utilizing information in relational database is much more difficult than Semantic Web.

6 Conclusion

The MAISS Music and Audio Ontology is an OWL-based application. As the subject area, that is music: artists, albums and tracks etc -- has so many competing requirements that a standalone format would not capture them all or would lead to trying to describe these requirements in a number of incompatible formats. By using OWL instead of RDF, the Music Ontology gains a powerfully extensible mechanism, allowing Music-Ontology-based descriptions to be mixed with definitions made in other OWL vocabulary. Moreover, by mixing inference engines like Jena with query language like RDQL, web ontology language like OWL, server pages like jsp, component architecture like Java Beans, and Servlet we designed and developed a powerful application which is able to accomplish something like magic.

MAISS resolved the problems associated with semantic information systems as follows:

1. Implementing multiuser transaction queries over files of ontology simultaneously have been solved by JavaBeans synchronized methods which had already been used in servlets.
2. Understanding the existing data within the ontology by the user which was solved by displaying subject, predicate, and object in browser.
3. Lack of existing web-based application to work with Semantic Web which has been solved by using the connection between Browser, Computation engine, Inference engine, and Ontology knowledge base.
4. Finally the difficulty of converting the Semantic Web formatting languages which has been solved by a converter program.

In MAISS, conversion of OWL to RDF is accomplished without selection of values by the user in that the selection of values by the users can be customized for future work. However, since in the present system the ontology is file-oriented therefore, whole contents of the ontology are loaded in RAM (Random Access Memory); this problem will be addressed in the future versions.

7 References

- OASF: Gayde, Erhan & Elci, Atilla, Ontology Appropriateness Score Finder, Proc. 4th FAE International Symposium, European University of Lefke, Gemikonađı, Lefke, TRNC 30 Nov. – 1 Dec. 2006, ISBN: 975-98897-1-4. pp: 407-410.
- IOMSP: Olgu, G. and Elci, A., Integrating Ontologies by Means of Semantic Partitioning, Canadian Semantic Web, Semantic Web and Beyond Series, Vol. 2, Koné, Mamadou Tadiou & Lemire, Daniel (Eds.), Springer (2006), 232 p., 20 illus., Hardcover, ISBN: 0-387-29815-0. pp: 121-134.
- DSSWS: Çelik, D. & Elci, A., "Discovery and Scoring of Semantic Web Services Based on Client Requirement(s) through a Semantic Search Agent", Proc. IEEE International Workshop on Engineering Semantic Agent Systems (ESAS 2006), pp: 273-278 in Vol. 2 of Proc 30th COMPSAC Annual International Computer Software & Applications Conference, 17-21 September 2006, Chicago, Illinois, USA, IEEE Computer Society Press, ISBN 0-7695-2655-1. pp: 273-278.
- MOS: Music Ontology Specification: Specification Document, <http://pingthesemanticweb.com/ontology/mo/>. 21 December 2006, Copyright © 2006-2007 by Zitgist LLC.
- MW: Music from Wikipedia: the free encyclopedia, <http://en.wikipedia.org/wiki/Music>. Accessed 25 September 2007.
- OWL: Web Ontology Language 2004, <http://www.w3.org/2004/OWL>.
- Jena: A Semantic Web Framework for Java, <http://jena.sourceforge.net>. Accessed 10 January 2007.
- RDF: Resource Description Framework 2004, <http://www.w3.org/RDF>.
- Jena Tutorial: A Programmer's Introduction to RDQL, <http://jena.sourceforge.net/tutorial/RDQL>. Andy Seaborne April 2002 Updated February 2004.
- JSP: Java Server Pages Technology: Copyright 1994-2006 Sun Microsystems, Inc. <http://java.sun.com/products/jsp>.
- Servlet: Java Servlet Technology: Copyright 1994 - 2006 Sun Microsystems, Inc. <http://java.sun.com/products/servlet>.
- JavaBeans: Desktop Java, JavaBeans: Copyright 1994 - 2006 Sun Microsystems, Inc. <http://java.sun.com/products/javabeans>.
- J2EE: Java EE at a Glance, Copyright Sun Microsystems, Inc. <http://java.sun.com/javaee>. J2EE 1.4.
- Algernon: Rule-Based Programming: Micheal Hewett, Monday, June 06, 2005. <http://algernon-j.sourceforge.net/>.
- MAISS: K. Bahreini and A. Elci (January 2007): "Music and Audio Information Search System using Semantic Web", Technical Report, Dept. Computer Engineering, EMU, TRNC.
- Giorgos Stamou and Stefanos Kollias, P. (2005): Multimedia Content and the Semantic Web methods, standards and tools. John Wiley & Sons Ltd.

Learning from Ontological Annotation: an Application of Formal Concept Analysis to Feature Construction in the Gene Ontology

Elma Akand¹Michael Bain¹Mark Temple²

¹ School of Computer Science and Engineering
Email: {akande,mike}@cse.unsw.edu.au

² School of Biotechnology and Biomolecular Sciences
Email: m.temple@unsw.edu.au

University of New South Wales,
Sydney, Australia 2052

Abstract

A key role for ontologies in bioinformatics is their use as a standardised, structured terminology, particularly to annotate the genes in a genome with functional and other properties. Since the output of many genome-scale experiments results in gene sets it is natural to ask if they share common function. A standard approach is to apply a statistical test for over-representation of ontological annotation, often within the Gene Ontology. In this paper we propose an alternative to the standard approach that avoids problems in over-representation analysis due to statistical dependencies between ontology categories. We use a feature construction approach to pre-process Gene Ontology annotation of gene sets and incorporate these features as input to a standard supervised machine learning algorithm. Our approach is shown to allow the straightforward use of an ontology in the context of data sourced from multiple experiments to learn a classifier predicting gene function as part of cellular response to an environmental stress.

1 Introduction

Ontologies are of growing importance in biomedical informatics and their uptake in this area may constitute one of the most successful applications to date of ontological engineering. Resources from open-source projects such as the Gene Ontology (GO) at www.geneontology.org are now nearly ubiquitous tools in bioinformatics (Bard & Rhee 2004). As of September, 2007 the original GO paper by Ashburner et al. (2000) had over 800 citations in in PubMed Central www.pubmedcentral.nih.gov.

A number of reasons can be identified for this success; for a review see Bada et al. (2004). In the current paper we are concerned with two aspects of the Gene Ontology that make it important for machine learning applications in bioinformatics.

First, the use of ontologies such as the GO provides a standard terminology for functional genomics – the objective of which is to describe the function of all genes in the genome of an organism – for example, in the analysis of gene expression data (Baldi & Hatfield 2002). Thus it is important for machine learning tools to work with such data. Second, the category definitions and hierarchical structure of the GO represent a “pre-Semantic Web” view of ontology,

where the goal of the representation was as a tool for human inspection rather than as a formal knowledge structure suitable for automated inference. There is a question as to how the structure of such a resource can be handled correctly by automated systems.

We are in the process of adding machine learning to a yeast data set analysis tool www.yeastinformatics.org, discussed below in Section 5. Therefore a key task is to allow the use of GO annotation of yeast genes in the preparation of training sets for standard machine learning tools.

In previous work (Bain 2002) we developed a feature construction approach using Formal Concept Analysis that demonstrated bias shift, leading to improved predictive accuracy with standard machine learning algorithms. Feature construction as a pre-processing step is a promising approach for handling non-vector based data as input to attribute vector-based machine learning methods. In this paper we investigate the applicability of this approach on GO annotation data.

The framework we develop allows an integrative analysis of heterogeneous functional genomics data based on standard machine learning tools.

2 Gene Ontology in Functional Genomics

The Gene Ontology is structured as an acyclic directed graph or DAG. Vertices or nodes are given a unique ID – a string of the form “GO:N” where N is a natural number – and a textual description intended to characterise some biological properties. Edges have two types, “is.a” or “part.of”. There are actually three sub-ontologies in the Gene Ontology, referred to as Molecular Function (MF), Cellular Component (CC) and Biological Process (BP). See www.geneontology.org for full details.

For most biologists the key role of the Gene Ontology is the use of the textual descriptions as a standardized vocabulary to annotate genes in terms of their function (MF), location in the cell (CC) and involvement in biochemical pathways (BP). Many genes are annotated with multiple descriptions. These descriptions, each with its unique ID and place in the GO DAG, are usually referred to as GO terms or categories (used synonymously) or simply nodes. For example, the gene known as CDC28 in the yeast *Saccharomyces cerevisiae* is annotated ¹ to one MF category GO:0004693 “cyclin-dependent protein kinase activity”, two CC categories GO:0005634 “nucleus” and GO:0005737 “cytoplasm” and eight BP categories including GO:0000082 “G1/S transition of mitotic cell cycle” and GO:0040020 “regulation of meiosis”. Shown in Figure 1 is part of the path to root of the BP ontology from GO:0000082.

Copyright ©2007, Australian Computer Society, Inc. This paper appeared at the 3rd Australasian Ontology Workshop (AOW-07), Gold Coast, Queensland, Australia. Conferences in Research and Practice in Information Technology, Vol. 85. Editors, Thomas Meyer and Abhaya C. Nayak. Reproduction for academic, not-for profit purposes permitted provided this text is included.

¹db.yeastgenome.org, accessed 28/9/2007.

```

1 %biological_process ; GO:0008150
2 %cellular process ; GO:0009987
3 %cell cycle ; GO:0007049 ; synonym:cell-division cycle
4 %mitotic cell cycle ; GO:0000278
5 <interphase of mitotic cell cycle ; GO:0051329 % interphase ; GO:0051325
6 <G1/S transition of mitotic cell cycle ; GO:0000082 % cell cycle process ; GO:0022402
7 %cell cycle process ; GO:0022402 < cell cycle ; GO:0007049
8 %cell cycle phase ; GO:0022403
9 %interphase ; GO:0051325 ; synonym:karyostasis ; synonym:resting phase

```

Figure 1: A fragment of the annotation path for CDC28 in the Biological Process ontology in the (now deprecated) GO Flat File format. Indentation on a line denotes refinement of a parent term on the previous line by the child term; the edge type is denoted by a ‘%’ for “is_a” or ‘<’ for “part_of”. Note the multiple inheritances on lines 5, 6 and 7. CDC28 is annotated to the term on line 6.

Since the initiation of the Gene Ontology project many groups have developed specialised GO software tools to handle both the ontology itself and data (typically sets of genes) annotated with GO terms. In the former category are tools such as ontology editors while the second contains mainly tools designed for *over-representation* or “enrichment” analysis, i.e., estimating the statistical significance of finding a set of genes annotated to a GO term.

2.1 Over-representation analysis

A common setting for over-representation analysis is computing P -values using a particular statistical test of gene sets resulting from a genome-wide high-throughput assay, typically a gene expression microarray experiment (Baldi & Hatfield 2002). In this context most tools adopt a standard probabilistic model for the number of genes that would be found by chance to be annotated to the particular GO category of interest. Typical choices for such models, and hence significance tests, include the hypergeometric or binomial distributions, or the χ^2 or Fisher’s exact tests (P. Khatri and S. Drăghici 2005).

A typical application of the hypergeometric distribution is the following, where we have a set of genes annotated to a particular GO term and we are interested in knowing the probability of finding that number of genes thus annotated simply by chance². We assume a “background distribution” of genes, typically the total number of genes in the genome with GO annotations, or the total number of genes in the experiment. This number is n . Of this set of genes, the size of the subset annotated to the GO term of interest is $m \leq n$. In the results of the experiment the size of our set of genes is s and the number of genes in that set annotated to our term of interest is r . The probability of finding by chance r genes from s thus annotated is given by the hypergeometric distribution

$$P(r, s, m, n) = 1 - \sum_{i=0}^{r-1} \frac{\binom{m}{i} \binom{n-m}{s-i}}{\binom{n}{s}} \quad (1)$$

2.1.1 Optimistic bias in probability estimates

A well known problem in the analysis of GO annotation using a probabilistic model to estimate statistical significance is that of *multiple testing*, in this case, where each GO term is tested separately using Equation 1. As the number of applications of a statistical

²This treatment follows that of Boyle et al. (2004), although we have corrected an error in the formula they present

test on a data set increases so the probability of obtaining an apparently significant result (P -value below the selected threshold) increases. This is usually allowed for by essentially lowering the effective significance threshold based on the number of tests by using the Bonferroni correction or alternatives (Boyle et al. (2004)).

In the case of over-representation analysis of GO annotation the issue turns out to be more complicated. The Bonferroni correction is usually thought of as being a conservative correction, i.e., the effective significance threshold is lowered more than necessary. However, Boyle et al. (2004) report that in their experiments the Bonferroni correction is not conservative enough, leading to an optimistic bias in estimating GO categories as statistically significant annotation for gene sets.

A simple qualitative argument can be developed to suggest why this is the case, and it has important consequences for the use of ontological annotation in over-representation analysis. The Gene Ontology, like many ontologies, is a generalisation hierarchy. This means that any object annotated to a GO term is also implicitly annotated to all of its ancestor (i.e., more general) terms. Just by considering the relation of a node to its parents, as in the following table, we can see the effect of this in terms of multiple testing.

No. of genes annotated	in Total	in Sample
to Parent GO term	$\geq m$	$\geq r$
to Child GO term	m	r

Using the notation from Equation 1 this table shows that the hierarchical structure of the Gene Ontology implies a dependency between the total number m out of n genes annotated to a GO term and the number ($\geq m$) annotated to any of its parents and hence all of its ancestors. The number of genes r in any sample of size s annotated to a GO term and its parents ($\geq r$) shows a similar dependency relation.

However, the Bonferroni correction assumes that each statistical test applied to the outcomes of an experiment (here, each set of genes annotated to a GO term) is independent. This can be expressed as an assumption that the values for r and m occurring in one statistical test have no relation to the values in any other test (s and n are fixed for any particular experiment). In the case of GO terms, as seen in the table, this is clearly incorrect. Once we have a GO term T annotating r genes, the probability of having other terms (i.e., the ancestors of T) annotating $\geq r$ genes is increased. A similar argument applies in the case of values of m .

In order to deal with this bias Boyle et al. (2004) implemented an alternative correction factor based on randomly sampling s genes from the background

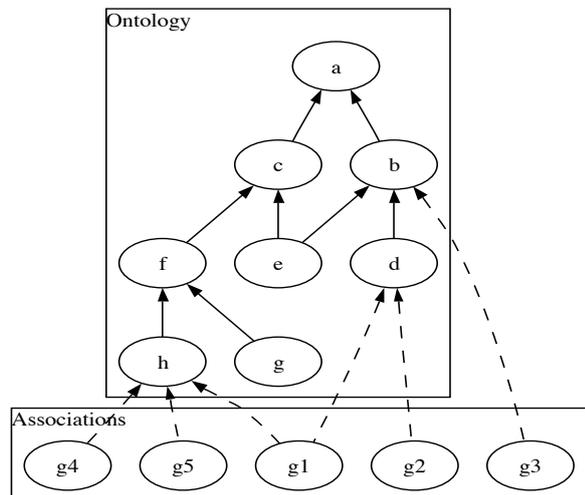


Figure 2: Example of a DAG-structured ontology with associated objects annotated to its terms.

set of n and applying Equation 1 to each GO term annotating any of this set. Repeating this procedure 1000 times gives, for each GO term, the proportion of apparently significant gene sets under the null hypothesis of random selection, which may then be used as an adjusted P -value. Although this way of computing a correction for P -values avoids independence assumptions it took three orders of magnitude longer to compute.

2.1.2 Coverage matrix

An approach developed by Carey (2004) accounts for GO structure in developing an information measure for GO annotation. In this approach GO edge types are ignored and edges are regarded as instances of the single relation $refines(C, P)$ where C and P are child and parent nodes.

Carey introduces the idea of an *object-ontology complex* in which the refinement relation from the GO DAG is represented as a binary (0,1) matrix Γ . Γ is a square matrix $V \times V$ where V is the number of terms in the ontology. For two terms T_i, T_j in the ontology, $\Gamma_{ij} = 1$ if $refines(T_i, T_j)$ for $i > j$, otherwise $\Gamma_{ij} = 0$. Matrix powers Γ^k represent k -step refinements of ontology terms.

A second matrix M maps P objects (here genes) to V ontology terms. In accordance with GO annotation policy it is assumed that genes are annotated to the most specific term. All 1-step refinements of the object annotations can then be computed as $C_1 = M\Gamma$. This is generalised using the idea of coverage, where a term covers an object if that term or any refinement of it is associated with the object via the matrix M . The binary coverage matrix C ($P \times V$) contains all such covers. Terms to which a gene is annotated are referred to as the “associated terms” or simply associations for the gene.

The coverage matrix can be used to calculate the probability of a term in the context of a specific object-ontology complex, i.e., a specific set of genes and their annotation in the Gene Ontology. The sum of the column i for a term T_i is n_i , i.e., the number of genes annotated to T_i or one of its refinements. The probability of that term appearing in the annotation of the gene set is $P(T_i) = \frac{n_i}{n}$ where n is the number of occurrences of the most frequent term in the annotation, typically the root node of the ontology.

Carey proposes an information-based similarity

	a	b	c	d	e	f	g	h
g1	1	1	1	1	0	1	0	1
g2	1	1	0	1	0	0	0	0
g3	1	1	0	0	0	0	0	0
g4	1	0	1	0	0	1	0	1
g5	1	0	1	0	0	1	0	1

Table 1: Coverage matrix for the example in Figure 2.

measure between terms. The information (in bits) of term T_i is $-\log_2 P(T_i)$. However, to the best of our knowledge no statistical tests for over-representation analysis based on the coverage matrix approach have been developed.

3 Ontological annotation for machine learning

Over-representation analysis is a commonly used approach that fits well the paradigm of exploratory data analysis. This is appropriate when the purpose of a biological experiment is “data-driven” rather than “hypothesis-driven”, e.g., to group together similarly behaving genes in a microarray experiment by clustering expression profiles (Baldi & Hatfield 2002). However, other experimental designs can be used that lead instead to results in which the data is divided into two or more groups. Then the task is to find a model or “hypothesis” that is the “best fit” to the data according to some criterion. In statistics this is known as discriminant analysis and in machine learning as supervised or classifier learning (Alpaydin 2004).

If the experimental setting is appropriate then discriminant analysis can have advantages compared to over-representation analysis. For example, we can mention two problems in functional genomics experiments with the approach of developing a probabilistic model then estimating statistical significance. First, as we have discussed in Section 2.1.1 simple probabilistic models may fail to take into account dependencies due to structure in a data set. Second, the difficulty of constructing correct probabilistic models increases rapidly with the number and diversity of data sets to be used for integrative analysis, where the goal is to combine multiple sources of data from different experiments to obtain a more complete picture of the operation of biological systems.

However, there are problems in the use of this kind of heterogeneous data with discriminant analysis or classifier learning also. Algorithms of this type require example data in the form of fixed-width vectors of attribute values. Much of the data used, for example in the Yeastinformatics web site (see Section 5), is not in this format. In this paper we focus on Gene Ontology annotation data.

In order to handle this type of annotation as data for classifier learning there are two problems to be dealt with, namely *multiple category annotation* and *multiple depth of annotation*.

The problem of multiple category annotation is that a given gene may have several different functions in a cell, and it may be found in several cellular processes and in different locations. Therefore, any gene is annotated with *all* of the categories with which it has been associated in the published scientific literature. In any particular experimental setting, however, only a subset of the known annotations of a gene will be relevant. This is known as a *multi-instance* problem. For propositional machine learning algorithms this is not an easy problem to solve.

The problem of multiple depth of annotation arises from the hierarchical arrangement of ontology categories and the way in which these are used to an-

notate genes. For example, two genes may have a related function, but are annotated at different levels of *generality*. Unfortunately, to a learning algorithm this relationship is not apparent; the categories are different. The learning algorithm could be modified to deal with the concept hierarchy. However, this is not straightforward and would have to be done over again for each learning algorithm to be used, which is impractical.

A solution to both problems may be provided by extending our previous results (Bain 2002) on *feature construction*. The idea is to pre-process the data containing multiple category and multiple depth annotation and use properties of the probability distribution on the annotation categories to generate new intermediate features which are then applied to the examples. These constructed features are then selected by the learning algorithm based on their utility in forming accurate models to predict the class of the examples.

3.1 Feature construction

Another way of viewing the coverage matrix of Section 2.1.2 is in terms of graph theory. It represents the *induced graph* for a set of genes and their associations with respect to the DAG structure of the Gene Ontology. Non-zero entries on each row denote all terms in the set of paths from the associated terms for that gene to the root node of the ontology.

The coverage matrix itself is a bipartite graph since it denotes a set of edges between genes and GO terms. It can also be represented as a “cross table” or formal context in the framework of formal concept analysis (Ganter & Wille 1999).

3.1.1 Formal concept analysis

Detailed coverage of Formal Concept Analysis (FCA) is in (Ganter & Wille 1999). In this section we follow the treatments of (Godin & Missaoui 1994, Carpineto & Romano 1993) since they are more oriented towards machine learning. However, some naming and other conventions have been changed.

Definition 1 Formal context A formal context is a triple $\langle \mathcal{D}, \mathcal{O}, \mathcal{R} \rangle$. \mathcal{D} is a set of descriptors (attributes), \mathcal{O} is a set of objects and \mathcal{R} is a binary relation such that $\mathcal{R} \subseteq \mathcal{D} \times \mathcal{O}$.

The notation $\langle x, y \rangle \in \mathcal{R}$ or alternatively $x\mathcal{R}y$ is used to express the fact that a descriptor $x \in \mathcal{D}$ is a property of an object $y \in \mathcal{O}$.

Definition 2 Formal concept A formal concept is an ordered pair of sets, written $\langle X, Y \rangle$, where $X \subseteq \mathcal{D}$ and $Y \subseteq \mathcal{O}$. Each pair must be complete with respect to \mathcal{R} , which means that $X' = Y$ and $Y' = X$, where $X' = \{y \in \mathcal{O} | \forall x \in X, x\mathcal{R}y\}$ and $Y' = \{x \in \mathcal{D} | \forall y \in Y, x\mathcal{R}y\}$.

The set of descriptors of a formal concept is called its intent, while the set of objects of a formal concept is called its extent. For a set of descriptors $X \subseteq \mathcal{D}$, X is the intent of a formal concept if and only if $X'' = X$, by composition of the $'$ operator from Definition 2. A dual condition holds for the extent of a formal concept. This means that any formal concept can be uniquely identified by either its intent or its extent alone. Intuitively, the intent corresponds to a kind of maximally specific description of all the objects in the extent.

The correspondence between intent and extent of complete concepts is a Galois connection between the power set $\mathcal{P}(\mathcal{D})$ of the set of descriptors and the power set $\mathcal{P}(\mathcal{O})$ of the set of objects. The Galois lattice \mathcal{L} for the binary relation is the set of all complete pairs of

intents and extents, with the following partial order. Given two concepts $N_1 = \langle X_1, Y_1 \rangle$ and $N_2 = \langle X_2, Y_2 \rangle$, $N_1 \leq N_2 \leftrightarrow X_1 \supseteq X_2$. The dual nature of the Galois connection means we have the equivalent relationship $N_1 \leq N_2 \leftrightarrow Y_1 \subseteq Y_2$.

The formal context $\langle \mathcal{D}, \mathcal{O}, \mathcal{R} \rangle$ together with \leq define an ordered set which gives rise to a complete lattice. The following version of a theorem from (Godin & Missaoui 1994) characterizes concept lattices.

Theorem 3 Fundamental theorem on concept lattices (Godin & Missaoui 1994) Let $\langle \mathcal{D}, \mathcal{O}, \mathcal{R} \rangle$ be a formal context. Then $\langle \mathcal{L}; \leq \rangle$ is a complete lattice³ for which the least upper bound (*Sup*) and greatest lower bound (*Inf*) are given by

$$\begin{aligned} \text{Sup}_{j \in J} (X_j, Y_j) &= \langle \bigcap_{j \in J} X_j, (\bigcup_{j \in J} Y_j)'' \rangle \\ \text{Inf}_{j \in J} (X_j, Y_j) &= \langle (\bigcup_{j \in J} X_j)'', \bigcap_{j \in J} Y_j \rangle \end{aligned}$$

Since we are concerned with concepts formed from sets of descriptors, the partial order as well as *Sup* and *Inf* definitions are given so as to relate to lattices in machine learning rather than that which is typical in formal concept analysis. That is, the supremum *Sup* of all nodes in the lattice in the “most general” or top (\top) node and the infimum *Inf* is the “most specific” or bottom (\perp).

3.1.2 Feature construction from concept lattices

Treating the coverage matrix as a formal context (Definition 1) where genes are objects and GO terms are descriptors enables the construction of concept lattices in which the formal concepts (Definition 2) contain sets of GO terms that group together sets of genes. The terms shared by such groups of genes indicates the biological properties that they have in common.

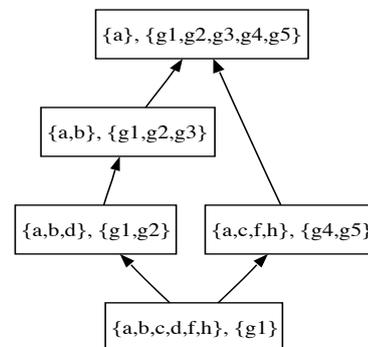


Figure 3: Concept lattice for the coverage matrix of Table 1.

In our previous work (Bain 2002) we investigated the use of concept lattices for both unsupervised and supervised learning. Both cases required the use of an information-based measure on formal concepts, similar to that of Carey (2004) discussed in Section 2.1.2. While both approaches are based on probability, our measure was motivated by the *compressibility* or algorithmic complexity (Chaitin 1987) of a concept. Compressibility of structured data objects, such as strings in a formal language or, as in this case formal concepts in a lattice, is inversely related to the probability of finding such objects by chance.

³Given a non-empty ordered set P, if for all $S \subset P$ there exists a least upper bound and a greatest lower bound then P is a complete lattice.

This approach was later used by us for ontology learning, essentially by extracting concepts from a concept lattice and combining them in a structured (propositional) logic program. However, while this is suitable for unsupervised learning it ignores the distribution of classes within the set of objects. Therefore, to use concepts in supervised learning we used a pseudo-MDL measure in which compressibility was combined with the entropy of the class-distribution of the examples in the intent of the concept. The intuition behind this is that concepts that will be useful in supervised learning will tend to have high compressibility and low class entropy; i.e., they will potentially lead to high accuracy classifiers.

For the current work where we are focused on the specific problem of handling Gene Ontology annotation in supervised learning we implemented a simpler approach to feature construction. This is described below in Section 4.3. Since we are using standard machine learning tools this also allows *feature selection* to be done after feature construction, as a training set pre-processing phase, where there are many powerful techniques available (e.g., in the Weka machine learning toolkit (Witten & Frank 2005)).

4 Case study: integrative analysis of cellular response to oxidative stress

A preliminary “proof of principle” experiment was carried out to test two aims of this work. First, we aimed to investigate whether a supervised learning approach using a standard machine learning algorithm was suitable to perform an *integrative* analysis of high-throughput results from multiple molecular biology experiments. If this was successful, a second aim was to employ a feature construction approach as described above to add Gene Ontology annotation to the data set constructed in the first step.

4.1 Biological background: cellular network response to stress

The number of organisms for which the complete genome (DNA sequence) is available continues to grow at an increasing rate. Meanwhile there have also been major advances in laboratory techniques to analyse complex cellular processes. This has ushered in a new era of cell biology, termed systems biology, in which responsive phenotypes, i.e., the measurable characteristics of the organism in response to environmental or genetic perturbations, can be investigated genome-wide, i.e., by collecting data on the activity of all the organisms genes simultaneously. In this way we can investigate the cellular network response of the genes that give rise to an observed phenotype as the downstream effect of an external stimulus through signal transduction.

For example, when cells adapt to sudden changes in the environment, cellular network responses include the action of sets of transcription factors (proteins) to activate sets of genes involved in biochemical pathways. Such a responsive sub-network of the cell is referred to as the genetic regulatory network (GRN). The protein products of co-expressed genes in a GRN combine to form interacting molecular machines that produce a responsive cellular phenotype. This responsive sub-network is partly described by the protein-protein interaction (PPI) network. In turn, proteins act to regulate cellular metabolism in pathways of biochemical reactions and, by subtle feedback mechanisms, their own GRNs and PPI networks.

The bakers and brewers yeast *Saccharomyces cerevisiae* is a key model organism for systems biology, due to the ease with which genetic manipulation can be carried out. Virtually all areas of cell biology have benefited from the use of yeast as a model organism to

study processes and pathways relevant to higher eukaryotes. Importantly, many fundamental processes in yeast are conserved through to humans. Data describing yeast cellular network responses are derived from high-throughput genome-wide experimental techniques; the development of which continues unabated. However, although a decade has passed since the sequencing of the complete yeast genome, fewer than 66% of yeast genes have a known molecular function. Even for those with a designated function this often denotes only part of their likely cellular role. Yet yeast is one of the most intensively studied organisms, with a relatively small genome. A key reason that knowledge on gene function is not greater is that the computational techniques and tools that will provide biologists with systematic ways to integrate the data resources and generate hypotheses about the function of cellular networks are not yet in place.

4.2 Experiment 1: integrating heterogeneous genome-wide data in supervised learning

As an example application of this problem of integration of data sets on yeast genes, we downloaded two sets of data on the same 92 genes. These two data sets give different “snapshots” of the yeast cellular network response to the addition to the cell’s growth environment of hydrogen peroxide. This environmental oxidant places the yeast cells under “oxidative stress”, causing the suppression of some normal functioning and the activation of cellular defence mechanisms. This provides a valuable experimental tool for studying cellular responses to environmental stress.

4.2.1 Classification task

In the paper by Godon et al. (1998) the authors identified 56 proteins whose synthesis was stimulated and 36 that were repressed under oxidative stress caused by exposure of yeast cells to hydrogen peroxide. This was a proteomics study, i.e., the results obtained reflected changes in the total composition of proteins in the cell using comparative two-dimensional gel electrophoresis. There is a relationship between protein synthesis, as observed in this study, and gene expression. For example, Godon et al. (1998) noted that the alterations they observed in the expression of proteins in response to hydrogen peroxide would be likely to involve a transcriptional component. In particular, the observed genomic response to oxidative stress strongly suggested an element of transcriptional control.

Accordingly, we designed a classification task to predict the protein response observed in the Godon et al. (1998) data in terms of attributes involving transcriptional response to hydrogen peroxide.

4.2.2 Attributes

In the paper by Causton et al. (2001) microarray data was collected on the cellular network response by yeast to a number of environmental stresses, including hydrogen peroxide. In contrast to the proteomics study above, this data was on the transcriptional response in terms of mRNA levels in the cell observed over a period of around 2 hours. This reflects genes that are “turned on” or “turned off” in response to the addition of hydrogen peroxide to the cellular environment. Data comprise a time series, with mRNA levels recorded at 10, 20, 40, 60 and 120 minutes following the initial exposure to the oxidant.

As a first step, we included the Causton et al. (2001) data along with a number of other attributes from the Yeastinformatics database (see Section 5 below). These include Gene Ontology cat-

egories (note: *without* the feature construction approach described in Section 4.3 below) and various other data types, described below in Section 5.

For comparison, we also constructed a training set with only the Causton et al. (2001) time series microarray data.

4.2.3 Results

Since this was a preliminary study, and we wanted to focus on the effects of representation change rather than the effects of choice of learning algorithm, we limited our attention to a single algorithm, the Weka implementation (Witten & Frank 2005) of the C4.5 decision-tree induction system (Quinlan 1993), called J48. For these experiments and those in Section 4.3 we used the default parameters for J48 and ran 10-fold cross-validation to obtain a mean predictive accuracy.

Decision tree learning in the first step gave a predictive accuracy of 83%. Given the fact that this representation using essentially raw data from the Yeast-informatics database, this was an acceptable result. However, using the Causton et al. (2001) data *alone* to predict protein induction/repression actually gave better results. The predictive accuracy rose to 86% correct, although with a slightly larger tree size (7 leaves, 13 nodes). This is interesting since it indicates that simply adding large numbers of attributes to the problem can degrade accuracy. In this case, feature selection did not help, since restricting the attribute set to the best ranked features using an information-gain metric did not improve accuracy.

In summary, these results confirmed the hypothesis of Godon et al. (1998) that transcription parallels protein expression. In doing so they confirm our hypothesis that supervised machine learning is a suitable candidate approach for integrative analysis of systems-level biological data. A standard machine learning algorithm, given no prior biological knowledge, was able to discover the relation of transcription to protein expression from data on cellular response to oxidative stress.

4.3 Experiment 2: adding GO features to the supervised learning problem

4.3.1 Classification task

The classification task was the same as that of Experiment 1.

4.3.2 Attributes

The set of attributes was limited to the Causton et al. (2001) data used in Experiment 1 plus Gene Ontology data pre-processed by feature construction based on Formal Concept Analysis as outlined above in Section 3.1.

However, this feature construction method focused on selecting *discriminative* concepts; in this case, those discriminating protein induction from protein repression. The procedure to construct features from a concept lattice for use in supervised learning was as follows.

1) for each of the genes in the set of 92 generate the gene's GO coverage as described in Section 2.1.2. For the example of Figure 2 this gives a set of "objects", i.e., genes:

g1 ← a, b, c, d, f, h.
g2 ← a, b, d.
g3 ← a, b.
g4 ← a, c, f, h.
g5 ← a, c, f, h.

2) construct a concept lattice \mathcal{L} from the objects shown as clauses in step 1).

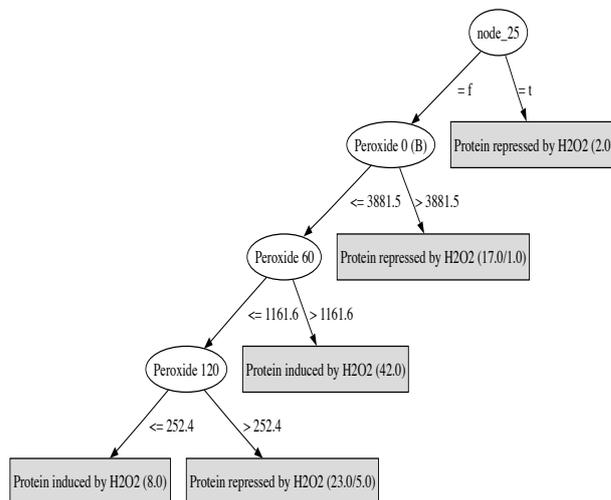


Figure 4: A decision tree for protein induction repression learned with Gene Ontology features. Ovals are attribute tests ("Peroxide t " means microarray data at time t), classifications are at leaves. See text for details.

3) for each formal concept in \mathcal{L} with extent containing ≥ 2 objects, evaluate the class distribution of the objects in the concept.

4) sort the concepts identified at step 3) in decreasing order of predictive accuracy for the majority class of objects in the concept.

5) select the top k concepts in the order identified at step 4), or simply all of those with accuracy above that of the frequency of the majority class (in this case, > 0.5).

6) construct a table containing, for each gene, a row noting whether, for each of the concepts identified at step 5) the gene is in the concept (i.e., feature is true) or not (feature is false).

7) join the table constructed at step 6) with the class values from the data of Godon et al. (1998) and the microarray data attributes of Causton et al. (2001) to form the training set for supervised machine learning.

This procedure is actually much more efficient than that of (Bain 2002) since it only involves constructing the concept lattice; once this is done, concepts can be evaluated quickly, and there is no expensive lattice revision as in the earlier method.

4.3.3 Results

With the addition of features from the molecular function ontology in this experiment the predictive accuracy of decision tree learning showed a slight increase to 87% correct, although with a slightly smaller tree size (5 leaves, 9 nodes), compared to the use of the Causton et al. (2001) microarray data alone. The tree learned is shown in Figure 4. In this tree the feature "node_25" stands for the following set of molecular function GO terms:

```
%molecular_function ; GO:0003674
%catalytic activity ; GO:0003824
%transferase activity ; GO:0016740
%transferase activity, transferring
alkyl or aryl (other than
methyl) groups ; GO:0016765
%methionine adenosyltransferase
activity ; GO:0004478
```

The question of whether this has any biological significance in the context of this data set is left to further work.

We also investigated adding features from the other sub-ontologies, biological process and cellular component separately, in all pairwise combinations, and all together. However, predictive accuracy was not as high at around 79-80% for these combinations. Although pre-processing the data with feature selection (not attempted) would probably remove these features, it is interesting that the GO annotation does not seem to give much additional predictivity in this context. This is discussed further below. Taken together, the results provide validation of the approach of feature construction from GO annotation as used in supervised learning.

5 Yeastinformatics web site

High-throughput genome-wide analysis of *Saccharomyces cerevisiae* has enabled many biological attributes to be assigned to each yeast gene. These data attributes include protein-protein interactions (PPI), transcription factor binding location analysis and protein location data. In addition, for each gene or encoded protein there are many curated sources of information that describe an associated metabolic pathway or ontology (Ashburner, M. and the Gene Ontology Consortium 2000). These data represent an invaluable resource to identify the nature of a particular gene or to define the relationship between genes. Typically this is achieved through over-representation analysis of a geneset to identify significant clusters of genes belonging to each biological attribute, such as a gene ontology or metabolic pathway.

The problem of data analysis has become more complex and time consuming since typically research laboratories now perform in-house genome-wide experimentation to address niche research interests. Consequently, over-representation analysis needs to be performed for many interdependent genesets. Apart from the issues of over-representation analysis discussed above, this is a problem since many of the popular analysis tools process each geneset in isolation and multiple analyses must be manually collated. Typically, multiple genesets are generated from one or more microarray experiments or deletion library screens.

The Yeastinformatics website is an extension of an earlier project (ScDSAT (2005)) with the goal of data set curation and gene set analysis. The Yeastinformatics web tools allow one to simultaneously determine which categories are over-represented in multiple genesets and to determine their relative distribution to determine whether these are common or unique. In addition, the Yeastinformatics web pages contain modules to facilitate: (i) the annotation of over-represented sets; (ii) listing of all database attributes for a single gene; (iii) graphical display of all PPI for a single gene and its interactions between all connected partners; (iv) graphical display of PPI within one geneset or between two genesets, and (v) listing of all genes bound by pair-wise combinations of transcription factors.

Approximately 200,000 gene attributes, such as an interacting protein, bound transcription factors or metabolic pathway are stored in a MySQL database, and these data are used for over-representation analysis, gene annotation or, as in this paper, generation of data sets for machine learning. *P*-values for the over-representation analysis are calculated using the hypergeometric distribution with optional Bonferroni correction for multiple testing (Boyle et al. (2004)) and only those intersections that pass the filter are tabulated. Additionally, the tables may be filtered by the numbers of intersections or not filtered at all. Cal-

culuation of hypergeometric distribution requires that the number of genes in the genome be known, however, not all genes have been verified and many are hypothetical or dubious. To allow for this, users can select which genome features to include for the calculation.

Yeastinformatics is implemented as a group of open access dynamic web pages. Backend scripts are implemented using HTML, PHP, MySQL, Java, SVG and Ajax. Each tool runs in a separate tabbed page each with separate back-arrow/previous-page functions for navigation. We recommend the Firefox browser for rendering the yeastinformatics web pages. All PHP scripts, HTML code and MySQL database information is available for download at <http://cgi.cse.unsw.edu.au/~yeastinformatics/cgi-bin/download>.

The machine learning approach described in this paper is implemented as a stand-alone tool, designed to be accessed from within the Yeastinformatics web pages. A screen shot is in Figure 5. Shown is a page from which the user can select data sets from the Yeastinformatics database and run the GO ontology pre-processing described in this paper to generate training sets in Weka file format for machine learning. The class and other attributes available in the Yeastinformatics database are shown in the panel on the left and those selected are shown on the right. The GO annotation sources are selected in the small panel on the bottom left and the method for GO feature construction is run from the panel at the bottom right.

6 Discussion

The problem of bias in over-representation analysis of GO annotation is still under active research. For example, several proposals for dealing with this bias in estimates by taking into consideration the graph structure and resulting dependencies in the ontology have been recently proposed (Alexa, Rahnenfuhrer & Lengauer 2006, Grossmann, Bauer, Robinson & Vingron 2007). However, the more general problem of dealing with dependencies in the complex data types such as interaction networks in an integrative setting remains.

In the context of machine learning, an alternative to the pre-processing approach we have described in this paper is to build ontology-handling directly into the machine learning algorithm. This is a long-standing idea in machine learning; a recent approach was implemented by Zhang et al. (2005). They incorporated "attribute-value trees" into a decision tree learning system. However, for ontologies of the size and complexity of the Gene Ontology it is not clear how well this approach will scale. Additionally, building in ontology handling requires modifying each machine learning algorithm one wishes to use, whereas pre-processing the training data into a standard format can allow the use of any standard algorithm.

A potential problem with our approach lies in the use of Formal Concept Analysis as the basis for our feature construction approach. Since each concept in the lattice has a set of descriptors that is "closed" with respect to the objects in its extent, the features that can be constructed are, in a sense, maximally specific. However, this is a form of *inductive bias* (Mitchell 1997) that may not be appropriate. In particular, it is not clear that this is an appropriate bias for the often noisy data that gene sets constitute. Although, since more general concepts may also be included as features, this may not be a critical problem, it should be investigated as part of future work. There are also known issues with the scalability of Formal Concept Analysis, and this will also need to be investigated.

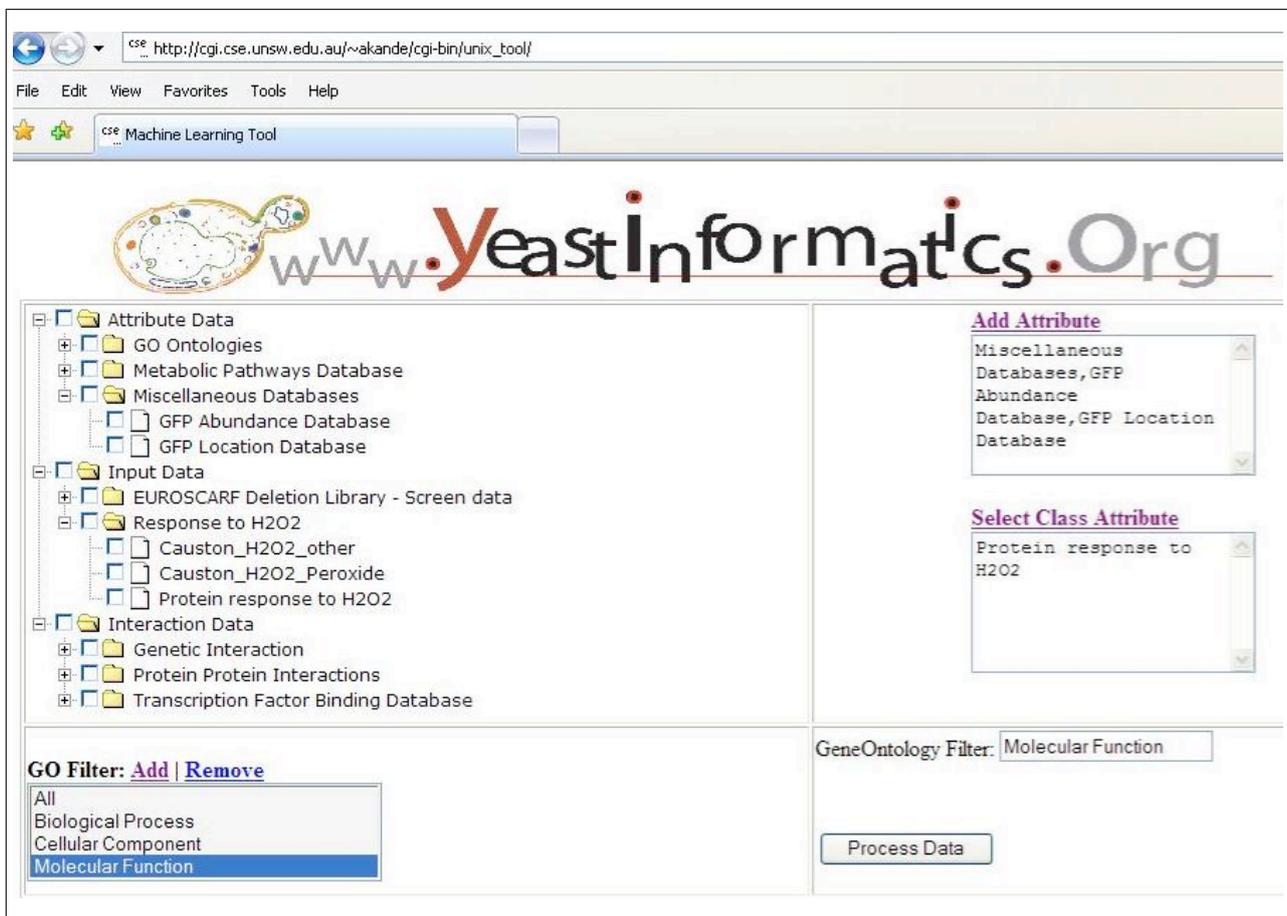


Figure 5: Yeastinformatics web tool for data set generation for integrative analysis.

In general, the reason GO annotation causes problems for vector-based machine learning is that it is a graph-based data representation. For example, the coverage matrix approach could be used directly as a set of attribute-vectors for such algorithms. This would lead, though, to very high-dimensional data sets (there are 23805 terms in the Gene Ontology of 27/9/2007, leading to a coverage matrix with up to 23805 columns). However, it is possible that kernel methods could be used, since they are often appropriate for high-dimensional data. We have chosen not to take this approach at since this stage we seek comprehensible models rather than “black-boxes” for use in the Yeastinformatics tools. Furthermore, with the addition of multiple graph-based data sources, such as protein-protein interactions, the data dimensionality could quickly rise to the order of 10^6 or even higher. Nonetheless, this could be investigated as part of further work.

A general property of graph-based data representations for machine learning is their sparseness when converted to vector format. However, other data sources in our Yeastinformatics database have this property, such as annotation of genes by the biological pathways in which they are involved, such as pathways in the KEGG database (Kanehisa & Goto 2002). A preliminary experiment using the feature construction method of Section 4.3 with KEGG pathway annotation replacing GO annotation resulted in successful incorporation of pathways features in the learned decision tree. We plan to investigate this further.

Wroe et al. (2003) was an early proposal to move the Gene Ontology to a description logic framework. This was motivated by the need for semantic analysis of the Gene Ontology by the use of description logic to enable validation, extension and classification tasks.

Currently multiple formats of GO are available for download, including OWL, MySQL and Prolog; we are using the latter two in our work.

Computing the coverage matrix can be done by bottom-up breadth-first traversal of the Gene Ontology from the gene associations. Then constructing a concept lattice by treating the coverage matrix as a formal context amounts to finding the minimal common graph paths for gene subsets. This appears to be related to the classification problem in description logics, and we plan to investigate possible connections as part of further work. Since GO is available in OWL format it makes sense to pursue this approach. Although the simple formalism of GO itself probably does not make this worthwhile, the possibility of applying this approach to ontologies in richer representations is one of our research goals.

However, it is not clear that description logics are always the best choice for ontology construction tasks; Stevens et al. (2007) found that using OWL for modelling complex biological knowledge was only partially successful due to limitations of the formalism.

7 Conclusions

Over-representation analysis applied to Gene Ontology annotation of gene sets obtained from high-throughput experiments was reviewed and the problem with bias resulting from dependencies due to the structure of the ontology was described.

We proposed an alternative approach that avoids the need for development of a probabilistic model by which significance can be assessed. By adopting a discriminant or supervised learning methodology we enable both the integration of heterogeneous data sources in a common “systems biology” framework

and the use of Gene Ontology annotation without relying on statistical tests to compute P -values.

Building on previous work we implemented a method for feature construction based on concept lattices to compute the common GO annotations for subsets of genes. Features selected from the concept lattice by a simple discriminative measure were then supplied to a decision tree learning algorithm. In an experimental application of the feature construction approach we found that GO annotation was incorporated into a learned tree together with attributes on microarray data to predict protein synthesis in response to oxidative stress.

As part of future work it is planned to extend the use of the approach to other data sets to enable a more detailed evaluation, both in application to GO annotation and other non-vector based attributes for machine learning.

Acknowledgements. Thanks to Rohan Williams for useful discussions, and to the anonymous reviewers for their helpful comments. This research was partly supported by a Faculty Research grant from the Faculty of Engineering, UNSW.

8 References

References

- Alexa, A., Rahnenfuhrer, J. & Lengauer, T. (2006), 'Improved Scoring of Functional Groups from Gene Expression Data by Decorrelating GO Graph Structure'. *Bioinformatics: Advanced Access*, published April 10 2006.
- Alpaydin, E. (2004), *Introduction to Machine Learning*, MIT Press, Cambridge, MA.
- Ashburner, M. and the Gene Ontology Consortium (2000), 'Gene Ontology: tool for the unification of biology', *Nature Genetics* **25**(1), 25–29.
- Bada, M., Stevens, R., Goble, C., Gil, Y., Ashburner, M., Blake, J. A., Cherry, J. M., Harris, M. & Lewis, S. (2004), 'A short study on the success of the Gene Ontology', *Web Semantics: Science, Services and Agents on the World Wide Web* **1**(2), 235–240.
- Bain, M. (2002), Structured Features from Concept Lattices for Unsupervised Learning and Classification, in B. McKay & J. Slaney, eds, 'AI 2002: Proc. of the 15th Australian Joint Conference on Artificial Intelligence', LNAI 2557, Springer, Berlin, pp. 557–568.
- Baldi, P. & Hatfield, W. (2002), *DNA Microarrays and Gene Expression*, Cambridge University Press, Cambridge, UK.
- Bard, J. & Rhee, S. (2004), 'Ontologies in biology: design, applications and future challenges', *Nature Reviews Genetics* **5**, 213–222.
- Boyle, E., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J. M. & Sherlock, G. (2004), 'GO::TermFinder – open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes', *Bioinformatics* **20**(18), 3710–3715.
- C. Godon and G. Lagniel and J. Lee and J.-M. Buhler and S. Kieffer and M. Perot and H. Boucherie and M. Toledano and J. Labarre (1998), 'The H202 Stimulon in *Saccharomyces cerevisiae*', *Journal of Biological Chemistry* **273**(34), 22480–22489.
- Carey, V. J. (2004), 'Ontology concepts and tools for statistical genomics', *Journal of Multivariate Analysis* **90**, 213–228.
- Carpineto, C. & Romano, G. (1993), GALOIS: An order-theoretic approach to conceptual clustering, in 'Proc. 10th Intl. Conf. on Machine Learning', Morgan Kaufmann, Los Altos, CA, pp. 33–40.
- Chaitin, G. (1987), *Information, Randomness and Incompleteness - Papers on Algorithmic Information Theory*, World Scientific Press, Singapore.
- Ganter, B. & Wille, R. (1999), *Formal Concept Analysis: Mathematical Foundations*, Springer, Berlin.
- Godin, R. & Missaoui, R. (1994), 'An incremental concept formation approach for learning from databases', *Theoretical Computer Science* **133**, 387–419.
- Grossmann, S., Bauer, S., Robinson, P. & Vingron, M. (2007), 'Improved Detection of Overrepresentation of Gene-Ontology Annotations with Parent-Child Analysis'. *Bioinformatics: Advanced Access*, published September 11 2007.
- H. Causton and B. Ren and S. Koh and C. Harbison and E. Kanin and E. Jennings and T. Lee and H. True and E. Lander and R. Young (2001), 'Remodeling of Yeast Genome Expression in Response to Environmental Changes', *Molecular Biology of the Cell* **12**, 323–337.
- Kanehisa, M. & Goto, S. (2002), KEGG for Computational Genomics, in T. Jiang, Y. Xu & M. Zhang, eds, 'Current Topics in Computational Molecular Biology', MIT Press, Cambridge, MA, pp. 301–315.
- Mitchell, T. (1997), *Machine Learning*, McGraw-Hill, New York.
- P. Khatri and S. Drăghici (2005), 'Ontological analysis of gene expression data: current tools, limitations, and open problems', *Bioinformatics* **21**(18), 3587–3595.
- Quinlan, J. R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA.
- R. Stevens and M. Aranguren and K. Wolstencroft and U. Sattler and N. Drummond and M. Horridge and A. Rector (2007), 'Using OWL to model biological knowledge', *International Journal of Human-Computer Studies* **65**, 583–594.
- Temple, M., Lail, A., Ratnakumar, A., Lam, L., Bain, M. & Dawes, I. (2005), 'ScDSAT: The *Saccharomyces cerevisiae* DataSet Analysis Tool'. *Yeast Genetics and Molecular Biology Meeting* (Bratislava).
- Witten, I. & Frank, E. (2005), *Data Mining (2nd Edn.)*, Morgan Kaufmann, San Francisco, California.
- Wroe, C., Stevens, R., Goble, C. & Ashburner, M. (2003), A Methodology to Migrate the Gene Ontology to a Description Logic Environment using DAML+OIL, in 'Proc. of the Pacific Symposium on Biocomputing', pp. 624–635.
- Zhang, J., Caragea, D. & Honavar, V. (2005), Learning Ontology-Aware Classifiers, in 'DS-2005: Proc. of the Discovery Science Conference', pp. 308–321.

Structure Based Semantic Measurement for Information Filtering Agents

Glenn Boardman Hongen Lu

Department of Computer Science and Computer Engineering

La Trobe University
Bundoora, Melbourne
VIC 3086, AUSTRALIA

Email: gcboardman@students.latrobe.edu.au, helu@cs.latrobe.edu.au

Abstract

With the volume of information on the Internet growing at an exponential rate, the needs of users to have their search results effectively filtered is increasingly important. A problem with most of the current search engines is that they only search on the specified keyword, which may be present in only a limited number of pages. This paper examines how a tree threshold function can be used in an information filtering agent (IFA) to extend the original keyword search to cover other related words within the domain, creating a keyword weighted semantic tree. The examination in this paper also considers how the metrics of the tree structure (shape, size, weights) influence the choice of related words for use in the extended search and what advantage this has over traditional methods. Further, that using a reduced word tree, which has been pruned using the tree pruning algorithm produces a significant increase in the number of profitable results for the user. Using these factors the analysis demonstrates equal accuracy to the benchmark comparison IFA but with increased efficiency and only a slight increase in execution time.

Keywords:

1 Introduction

As information on the Internet continues to grow at an exponential rate, the ability to search web pages and return meaningful results becomes a more daunting task. Search engines such as Google and Yahoo can return hundreds of thousands of web pages links with little certainty about whether they contain any information relevant to the user's area of interest (also known as the search domain). For instance a search for the word 'jaguar' returns 95.4 million results in Google and 39 million results in Yahoo with topics ranging from Jaguar motor cars, car clubs to mountain lions. In this situation, it is up to the user to painstakingly filter hundreds of results presented to them or to iteratively search within these results with additional keywords. In reality, most users probably only view the first few pages of results without proceeding further.

This problem arises because current search engines do not take the search domain into account. To alleviate this problem, one approach is to use an Information Filtering Agent (IFA) to automate the filtering process. An IFA not only analyzes the occurrences

and location of keywords within a document, but also analyzes the relationships between keywords. These relationships include the study and detection of related words from a search and how they may be used to further assist the user in their search.

In this paper, we propose an adaptive function to measure the semantic distance of keywords taking into account of domain knowledge and the shapes of ontologies. This function serves as an important role to improve the accuracy of information filtering agents. Experiments and analysis show promising results using this function in IFA.

2 Information Filtering Agent

Whilst it would be useful to provide a definition of an information filtering agent (IFA) in this section, it is difficult to give a generic definition as the nature of an agent is as varied as its implementations (Woolridge 2002). Each programmer has a different style of coding and a different view as to how the agent should operate. An ontology based web IFA begins with a data pre-processor designed to parse a dataset converting it to an understandable form (such as plain text) and removing all useless information such as stop words and html tags (Lau et al. 2001). The resulting dataset may be a database of web pages, or may be a set of search results from another search engine such as Google. The IFA then applies semantic knowledge and ontologies to a data post-processor in order to refine a dataset on behalf of the user (Sim 2004).

The refinement process in an IFA is often similar to that of a search engine. Keyword frequency and location based methods with ontology extensions are popular, as well as clustering based methods where a page is given a designated category and an appropriate domain is selected by either the system or the user (Hotho et al. 2001, Prabowo et al. 2002, Guandong Xu 2005).

The ontologies are used to represent a semantic link between two words. These relations may be represented in many ways, typically as a either a matrix or a tree structure. The most commonly used structure available is WordNet (Miller et al. 1990), which is discussed below, however some authors create their own ontological trees. Another online application that allows users to build their own ontology graphs is Ontolingua (University 2003). However, the focus of this paper will be on the WordNet network due to its broad availability and more extensive word listings.

It must be taken into account that a related word, whilst being similar, is not identical to the term specified. To counter this, words may be weighted to represent the strength of the bonds between them (Cesarano et al. 2003, Varelas et al. 2005). An example of a word tree around the word 'jaguar' is

Copyright ©2007, Australian Computer Society, Inc. This paper appeared at 3rd Australasian Ontology Workshop (AOW-07), Gold Coast, Queensland, Australia. Conferences in Research and Practice in Information Technology, Vol. 85. Editors, Thomas Meyer and Abhaya C. Nayak. Reproduction for academic, not-for profit purposes permitted provided this text is included.

shown in Figure 1 (note that the word tree is incomplete). How these numerical relationships are generated is the focus of this paper (Cesarano et al. 2003).

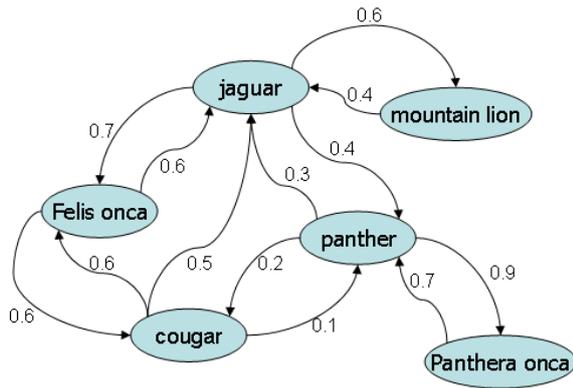


Figure 1: A sample weighted word tree

2.1 WordNet

Progressively developed by Princeton University since 1990, WordNet is a lexical reference system containing over 207,000 word-sense pairs, which represent lexical relations and concepts. WordNet utilizes a lexical matrix to represent the relationships between words as opposed to the semantic tree discussed earlier (Green et al. 2001). WordNet was designed for the implementation of search software, which makes it highly useful in the design and programming of an IFA. There are many applications of IFAs which continue to use WordNet as their semantic backbone due to its practical usability, and its lexical coverage.

3 Existing Approaches to IFA

Given the many different ways that an IFA can be conceived and implemented, it is virtually impossible to directly compare every approach taken. For this reason, we review the different implementation in each system and consider them independently.

3.1 Three Stage Information Filtering Agent

This IFA determines the relevance of web pages by utilizing three heuristics: detecting *evidence phrases* constructed from WordNet, counting the frequency of evidence phrases, and considering the nearness among keywords.

An evidence phrase is a semantic variation on the original search term(s). This approach utilizes all possible semantic variations of a keyword such as synonyms, hypernyms and hyponyms. Hypernyms and hyponyms are generalizations and specializations of a word respectively, for instance 'big cat' is a hypernym of 'jaguar' and 'baby jaguar' is a hyponym of 'jaguar'. Throughout this paper, these terms (synonyms, hypernyms and hyponyms) will be used frequently as they describe the most common semantic relations.

Evidence phrases may be constructed on either a whole word or partial word basis. By analyzing the full text of the search, the system has a better basis for domain analysis since more words may provide a more specific description as to the nature of the search. This is where a limitation of the system comes into effect. Short queries can produce irrelevant data as there is no context to base the search on. For instance a search for the word 'battery' may return results about electric batteries or it may return results concerning artillery. Given the search term, it

is impossible to tell the true nature of the search and both sets of results must be included. From evaluation of this work, this system appears to have several limitations. As above, by only taking one level of related words into account. The possible derivations of evidence phrases may lead to domain overlap and poor performance. However with the addition of an advanced scoring function, this type of agent could be an excellent platform for testing given the benchmark results supplied with the paper and details of operation.

3.2 Semantic Knowledge Base Filtering Agent

This approach uses a data post-processor to filter search results. However, it is further defined to be a hybrid post-processor using a semantic knowledge base to grade pages according to semantic similarity.

3.2.1 System architecture and algorithm

This system comprises multiple stages to refine the search results, from the raw data extraction (meta crawler) to the results display. The first stage is a search engine wrapper which accepts the users query and reshapes it to fit most common search engines such as Yahoo and Google. The results returned are parsed and links are extracted from the pages. The *submitter* is essentially a customized web browser that will search the extracted links. This data is passed to the *web spider agent*, which sorts and orders the pages into a storage area. Once the data is in storage, the *web page parser* extracts the links from the generic page, which are reliant on the same logical web site. If these links do not exist within the storage area, they are added to the link repository.

3.2.2 The semantic knowledge base

The core of this system is where the semantic relationships are developed and where concepts (or ontologies) are used to build a semantic network. The purpose of the network is to define a domain formalization representing a domain of objects and their relationships. The difference between this agent and others is how the graph weights are determined.

3.2.3 Probabilistic distance function

The weighted graph used to represent the linguistic ontologies has two components: nodes and paths. The nodes are the words and the paths are the interconnections or relationships between them. As the English language is complex, and words are not simply derived from each other, each node may have multiple paths to and from it. This is a useful feature as it can be utilized to construct the tree weights. By looking at the number of edges from point *a* to point *b* this agent uses heuristics in order to determine the appropriate weight for a given semantic relationship. This is a significant development in that this function is dynamic. Most other processes will use a static assignment of weights as previously discussed. One issue this eliminates is the need to check the kind of semantic relationship (if word '*a*' is a synonym, hypernym or hyponym of '*b*'). As there is a direct weighting to the next word on the semantic tree, the weight can be applied without the need for further calculations.

3.2.4 The data miner

This is the main agent of the system and its task is to grade pages in the data repository according to

their relevance and category. Operating in a similar way to a clustering algorithm, a page can be assigned to one or more categories based on its estimated appropriateness to a the user specified domain and the ontologies used. The data miner is a binary classifier, which means that in analysis that there is a specific domain which is to be classified and its complement. This agent may be broken into three sub-components: syntactic grader, semantic grader, and global grader. The syntactic grader evaluates the semantic and syntactic relevance of pages from search engines based on ontologies. One downfall of this process is that it favors highly ranked pages from each search engine. Each search engine is given a weight such that for n search engines, $\sum_{i=0}^n searchengine_i \cdot weight_i = 1$. The semantic grader uses ontologies to grade pages. These are considered to be a function of concepts where each concept is expressed by a word. The global grader is simply a linear combination of the other grading factors.

3.3 SHOE

This system displays the true diversity of potential solutions. It does not use a filtering agent to extract or derive semantic meaning from web pages; instead an ontology extension is applied directly to the HTML describing its contents, author and related information. There are previous works along the same line as this, HTML 2.0 (Berners-Lee & Conolly 1995) incorporates several weak concepts for semantic markup and newer versions of HTML extend this further. As the required semantic information is already available (WordNet), this system proposes the use of semantic searches such as “*find me all graduate students*” to identify relevant information. This approach overcomes some previous limitations with other ontology based IFAs, with its ability to analyze names within searches. For instance a search for someone named ‘Cook’ in Google will return not only people named Cook, but information about cook books and the like. Identifying names within traditional IFAs is a difficult process as there is no way to tell what portion of the query is the name and what is not.

4 Problem Statement

The problem with most current implementations is the lack of domain knowledge and inadequate word tree scoring functions. Ontology scores are usually either static or are specific to only a few words derived from a semantic tree. Whilst these methods are useful, there are limitations. Static values do not allow for the possible variance that topics can have and cannot compensate for topic ranges. Weighted semantic word trees are an excellent option as they can cater for topic variations, however the tree structures are usually manually designed and the ontology scores are determined and influenced by the programmer’s perception.

The primary focus of this research is to design and develop an IFA that improves upon existing manual structure designs. A previously untested method used in this paper is the processing of word tree structures using geometric analysis (eg shapes, dimensions, scope). Tree structures are used in many areas of study to represent many different things. However, there is one underlying construct between all tree structures; their variability in scalability, shape and other metrics. Given this fact there are an infinite number of factors for consideration when analyzing tree structure’s and a generic function to represent all of these metrics is essentially undefinable.

A word tree threshold function is needed to analyse the formed tree shapes and other attributes when a keyword is searched on the Internet and to determine the level of derived focus nodes for use when refining the search. The tree threshold function will use an existing semantic network such as WordNet (and a statically assigned semantic tree as discussed above) to take advantage of its extensive coverage of many topics and allowing focus to be placed on the scoring function, which is the critical element.

5 Adaptive Semantic Measurement for IFA

There are many approaches that may be taken to implement an IFA, utilizing different methodologies to achieve an outcome such as using static scoring or link analysis. In comparison to these implementations the IFA developed in this paper operates in a very different manner. Since previous approaches do not analyse the underlying structure and weights of the word trees, their methods of scoring web pages are not applicable to this IFA. To implement this IFA a method was developed to traverse the tree structure and to remodel it based on a set of threshold scores calculated to correspond to a particular branch height or level within the tree.

The proposed IFA works by generating a set of threshold weights derived from the structure of the tree generated by WordNet and the weights of this tree using the factors reflecting the domain. This threshold set is defined as $T_h = \{W_1, W_2, \dots, W_n\}$ where n is the maximum level of the tree structure. These thresholds are then used to prune the generated keyword tree (tree pruning agent), removing derived nodes (hypernyms and hyponyms) that will not contribute to the extended search. For a word to be included in the extended search, its weight must exceed the corresponding weight within the threshold set as determined by the threshold generator. If a node has a weight less than its corresponding threshold value, it is not related closely enough to the keyword to be of use within the next stage. For an example of a weighted tree, see Figure 2.

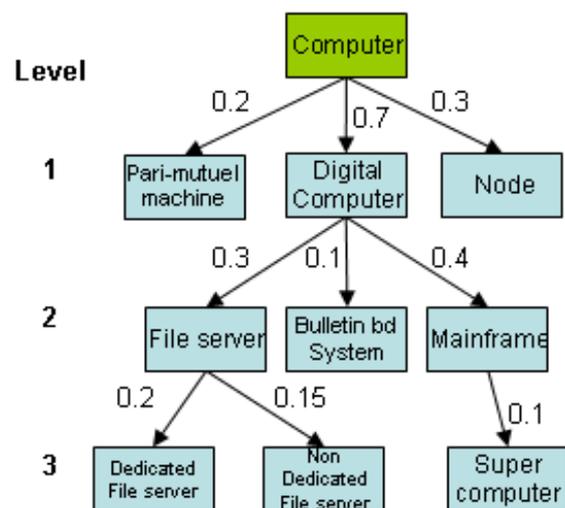


Figure 2: A segment of the word tree for “computer”, with weights

The method of generating the thresholds for hypernyms is different, however the operation of the remaining elements of the IFA is primarily the same. Once the threshold set has been calculated, any node which does not exceed its threshold level is removed. By its nature, a hypernym tree is much simpler than

a hyponym tree, and as such the threshold generator function for hypernyms is not as complex.

After the tree pruning algorithm has been executed the tree structure should contain only those words with weights that exceed their respective threshold levels, and therefore only the words that will usefully contribute to the user's search. From this point, the structure of the tree becomes irrelevant since all the remaining nodes have been established as being relevant to the original keyword. Therefore it is possible to flatten the tree structure to a list of words and weight ready for a page to be scored.

Scoring is a simple process, recursing through each node from the tree structure and counting the number of occurrences within the given document. Once this document has been scanned, its score is updated by adding the number of occurrences of the word to the original score, each multiplied by their respective weights.

6 Threshold Function Factors

This section covers the threshold generation function and the factors that have been taken into consideration, how they effect the overall operation of the IFA and whether they are profitable in refining search results.

6.1 Parameter Usage

Apart from the possible number of factors that could be addressed in the threshold function to evaluate its characteristics, there are also as many ways that these factors could be utilized within the function. The manner in which a factor is used may also have a significant effect on the outcome of the function and dictate its behavior and characteristics.

A base factor is a raw data statistic or set of values, which is added to the function to increase/decrease the threshold value accordingly. Base factors are useful as they provide a backbone for the threshold values. A scalar simply scales the backbone values to fit within a set of guidelines, or to alter particular set values to adjust that level's threshold.

6.2 Parameters Studied

6.2.1 Average Tree Weight Per Level

A problem with most of the factors used in this study is that they may be replicated easily with many different trees and many permutations of words. One tree structure may be geometrically identical (or similar) to another tree structure, generated from an entirely different word and domain. The problem is how to differentiate between these seemingly identical trees. Given this situation, a threshold value determined solely on the metrics (shape, size, scope) of the structure is too generic and is ultimately of little value without a unique aspect to reflect upon. The most unique aspect of each the word trees as examined are their weight schemes which are unique and does not apply to any other tree. Therefore, it is logical to use this feature to create a baseline for the threshold function from these weights. Although this is not a true metric of the tree structure, it does provide an accurate indication as to the nature of the structure.

As this function analyses the thresholds on a per level basis, all weights on a given level must be taken into account when creating the threshold. To achieve this, the threshold used as the baseline is the mean average of all weights on that level.

6.2.2 Tree Shapes

The shape of a word tree is the most direct and obvious metric to take into account when attempting to characterize the shape of a tree structure. There are a limited number of generic shapes that may be formed within a tree and some of these shapes are shown in Figure 3.

The tree shape factor could also be referred to as the characteristic decay factor as it predicts the likely decay and the scale of scores that should exist within a tree. For instance, the bell shaped tree should have a sharp decline of the weights as it approaches its lower levels, because there are an increasing number of topics (derived focus nodes) being generated, which are theoretically less relevant to the focus word and therefore have a lower weight. See Figure 3 for more tree shape examples and theoretical score decay graphs.

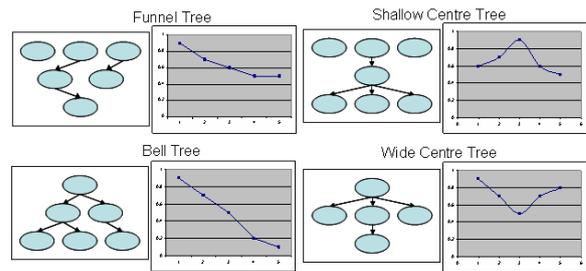


Figure 3: The differing tree shape types and score decay graphs

This theoretical decay due to the shape is useful as it should accurately approximate the behavior of derived focus nodes as the approach tree height H_n . This factor is generated from a quadratic function centered at $x = \frac{n}{2}$ and evaluated at each tree level.

The shape of a word tree is determined by cutting it into three sections according to its maximum height/depth. For instance, a word tree with six levels will have three sections: height zero to two, three to four and five to six. Three divisions were chosen because it provides a few highly definable shapes and due to the practical limitations of programming.

6.2.3 Average Tree Width and Height

This factor is intended to give scope to the scale of the word tree structure. As discussed previously, many different keywords can generate seemingly identical structures based purely on their shape. By analysing the average height and width of these structures the characteristics of the structure can be identified.

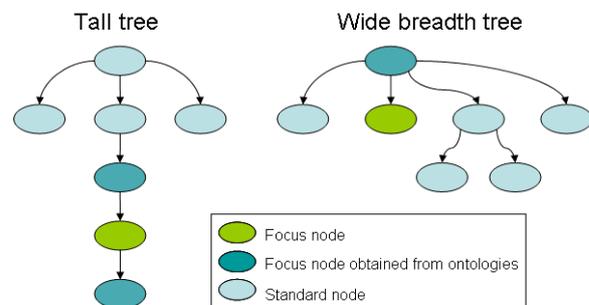


Figure 4: Example high and wide tree structures

The physical size of the tree may have a bearing on the threshold values since a very large tree contains many nodes but not all of these nodes are needed to extend the search criteria. A word tree with a shape,

average weighting and synset size that is identical to another tree may contain fewer words per level and therefore requires more nodes. Without knowing the dimension of the tree structure, the ability to determine its metrics and behavior is limited. The factors of average height and average width are defined as A_h and A_w respectively. See section 6.3 (surmised threshold function).

6.2.4 Individual Branch Heights/Depth

Much like the average tree width and height, this metric gives an accurate measure of the nature of the tree. Its main purpose is to identify certain individual branches that may form a particular aspect of the shape. The branch height function is intended to identify any particular branches that extend into the depth of the tree, which would be a very specific subtopic of the focus node, and therefore is more likely to contain useful information. Given this, it is more profitable to use this branch in the final search as it will return more useful and relevant results for the user.

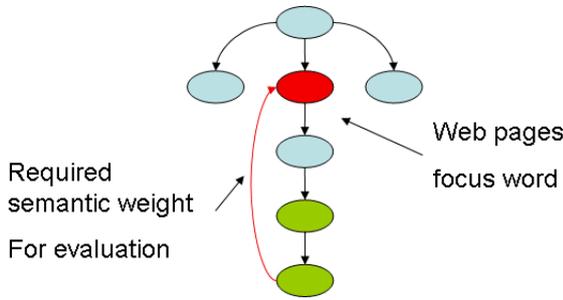


Figure 5: Example of branch height/depth

It is important to note the key difference between this metric and the average tree width and height metric. This factor takes an individual branch of the tree into consideration, not the overall level or width of the tree. The branch height/depth is defined as BH_i where i is the index of the branch in question.

6.2.5 User Input - Pruning

In many situations, external information may be as useful as the information generated internally by the tree threshold function. Therefore it is profitable to provide for user input to characterize the desired level of output and extent of the results. The user may also choose to prune the tree lightly, whereby the thresholds would be scaled accordingly to allow a greater number of derived focus nodes into the search.

In the IFA developed for this paper, this is achieved by using a scaling factor to adjust the threshold values per level. The options presented to the user are shown in Table 1.

Prune type	Scale value
Normal	0.9
Light	1.0
Brutal	0.8

Table 1: User input scaling values for tree pruning

This pruning factor is rather simple in implementation, but should have a substantial influence on the thresholds given that the scale of the threshold values are zero to one and a change of 80% can be very significant.

This factor is defined as $U_i = \{0.8|0.9|1.0\}$, providing three possible user selected values for input in

order to scale the threshold. see Section 6.3 (surmised threshold function).

6.2.6 Synonym Set (Synset) Size

Synonyms are an important consideration for developing a threshold function as they represent the strongest semantical link between the keyword, since a synonym is quite often interchangeable with the original word. An extensive synset such as in Figure 6 provides a very productive source of additional search words. A large synset like jaguar’s can indicate that there are many other words that could be used in place of the keyword and therefore, a more inclusive domain is likely to return more search results for the user.

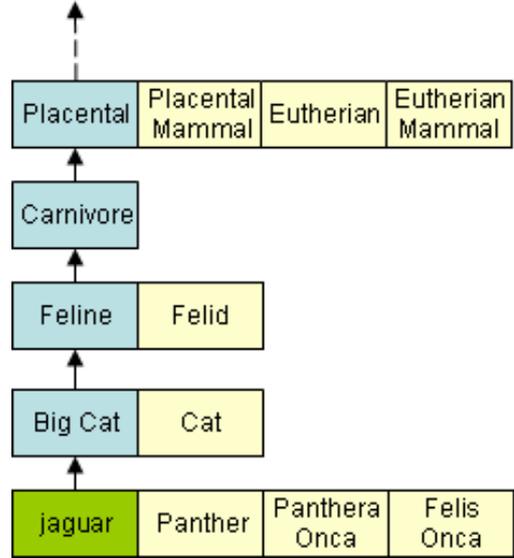


Figure 6: Part of the “jaguar” hypernym tree showing synsets

The synset size $SS_{n(i)}$ is used as a scaler in the surmised threshold function as it provides a primary analysis for the tree structure.

6.3 The Surmised Threshold Function

To achieve the optimal output from the threshold generator, it is important to incorporate the discussed factors in an appropriate way. The base of the threshold generation function centers around two factors being the average weight per level and the tree shapes. The average weight is the most important factor as it outlines the weighting of the word tree, but the tree shape is also important because it indicates the individuality of the topic. The formal definition of this function is as follows:

$$T_s = U_i \times \frac{\sum_{i=0}^{I-1} BH_i}{A_h \times length(BH)} \times \prod_{height=0}^{N-1} \left(\frac{AW_{height} + TS_{height}}{2} \times \frac{SS_{height}}{length(SS_{height})} \right)$$

The core of this function is based around the “Average branch height” and the “tree shape decay” factors which are taken as a 50/50 value. As discussed above, these two factors are used as base factors since they provide the most meaningful description of the tree shape. Given any tree height n , these two base factors are evaluated and then multiplied with the other factors to vary the thresholds generated.

6.4 Hypernyms

Not discussed in this section are the factors relating to the hypernym component of a word tree structure. By their nature, hypernym word trees are not as complex as hyponym word trees as their content tends towards very abstract words such as 'entity' and 'object' within very few levels of the root. Given this attribute, a complex hypernym function taking into consideration the many metrics of the tree, is not warranted. For a more detailed discussion of the hypernym threshold function, see Section 7.4.

7 Evaluation and Analysis

7.1 Evaluation Methods

The means of evaluating the threshold generation function are simple. Two tree structures were chosen, which have varying shapes and dimensions that were subjected to the following comparative evaluations:

- **Single word** - the keyword alone to gauge the content of the dataset for that domain.
- **Full tree, unitary weight (FT-UW)** - All nodes within the word tree (hyponym and hypernym) are given a weight of 1.0
- **Full tree, fully weighed (FT-FW)** - All nodes within the tree are included, except that they are given their full weighting.
- **Full tree, unitary weight, threshold cut (FT-UW-TC)** - Same as the unitary weight test, except that it uses a tree which has been trimmed with the threshold function.
- **Full tree, fully weight, threshold cut (FT-FW-TC)** - Same as the fully weighted test, except that it uses tree which has been trimmed with the threshold function.
- **Comparison function** - Use the comparison function described in Section 7.2 to assess the comparative correctness of the output.

Each test was run on the SQL data set in order to generate a results set. From this data a table has been populated and shows the generated page rank, page index and percentage of relevant documents returned by the page scoring agent (in contrast to the comparison function and human ranking). This testing scheme ensures that if there is a detectable difference in the threshold functions results, it can be properly identified.

7.2 Comparison Function

There have been very few studies that have examined semantic tree structures as a means of object reduction and information evaluation, which have published their methodologies of evaluation. Consequently no other studies have results that can be directly compared to the outcomes of this paper in order to evaluate the effectiveness of the developed IFA.

However, an effective algorithm for page evaluation is outlined in (Sim 2004) where static weights are assigned to hypernyms, hyponyms and synonyms. In other studies, this approach has been shown to improve the accuracy and quality of results and is therefore a good IFA for comparison with the IFA which is the basis for this paper. The results of the comparison are included in the following section.

7.3 Hyponym Evaluation

Given that the hyponym function is comprised of six components, finding a tuning point where all factors provide the greatest positive effect on the tree threshold function can become difficult. This is made more difficult by the changing nature of the word tree structures. For example, the tuning point for a balanced bell shaped word tree may be completely unprofitable for an elongated diamond shaped tree.

For this reason many of the factors used in the tree threshold function were chosen as either global or per level scalars. This has allowed the focus of this paper to remain on the two most important base factors, being the average weight per level and word tree shape (characteristic decay).

The first test conducted was to find the balance between the base factors that provided the most appropriate threshold values. Using a bias towards the average weight such as 60%-40% would yield a more accurate representation of the tree structures but was more likely to include unproductive nodes. Tests show that the average height per level is more likely to include both highly weighted nodes and mid weighted nodes. This is not the desired functionality of the tree threshold function because the aim is to only identify the nodes which have the highest possibility of being present in web pages. By moving the bias towards the characteristic decay (40%-60%) increases the level threshold at lower heights and relay the thresholds towards the top of the tree. This means that more nodes towards the top of the tree are included which, is as undesirable as the biasing of the thresholds towards the averages. In the three tree structures tested, the optimal balance between average weight and characteristic decay was 52%, 56% and 45% which indicates that a level bias provides optimal output. Thus the decision was made to set the balance at 50% average weight and characteristic decay. Whilst this did not provide optimal results for any given tree shape, it did provide the best average results.

Shown in Tables 2 and 3 are the results obtained in this study for the keywords 'computer' and 'car'. These tables show the scores and page indices obtained after the conduct of the tests. The relevance percentage listed at the bottom of each column was determined using the following method. When the resultant pages are identified by the output generation agent, the page is viewed and manually assigned a score between one and ten indicating the relation to the original keyword and its context (0 indicating no relevance and 10 indicating high relevance).

These results indicate that the use of a semantic tree significantly increases the accuracy of an IFA and in some situations identifying web pages that were undiscovered by single keyword searches. Figures 3 and 2 also show unexpected results. As expected, the worst result came from using only the original keyword, scoring 0% for 'computer' and 90% relevance for 'car'. This happens because not every document will mention the word 'car' or 'computer' however its content may be within the same domain, and therefore noticed when the extended search criteria is added.

The most interesting result is that a complete word tree with unity weight performed the worst of all, identifying only 85% and 80% of the relevant documents. This may be a counter intuitive result, but there is a possible explanation. Several identified pages were false positives due to the excess extended words which are not specific to the domain in question. The presence of these words within the document means that the page score was being falsely increased by a significant amount owing to the fact that these words had a score of 1.0 when they have

	Single Word		FT- UW		FT- FW		FT- UW- TC		FT- FW- TC		Comparison	
	Page ID	Score	Page ID	Score	Page ID	Score	Page ID	Score	Page ID	Score	Page ID	Score
Top ranked Pages	No results		86	95.00	86	40.25	86	95.00	86	34.80	86	38.00
			494	95.00	494	40.25	494	95.00	494	34.80	494	38.00
			157	29.00	97	7.20	97	18.00	97	7.20	97	7.20
			565	29.00	505	7.20	505	18.00	505	7.20	505	7.20
			277	20.00	251	6.30	202	13.00	251	5.40	202	5.20
			685	20.00	659	6.30	610	13.00	659	5.40	610	5.20
			97	18.00	261	5.81	121	12.00	202	5.20	121	4.80
			505	18.00	669	5.81	529	12.00	610	5.20	529	4.80
			202	17.00	202	5.22	251	10.00	121	4.80	251	4.00
			610	17.00	610	5.22	659	10.00	529	4.80	659	4.00
			121	12.00	121	4.80	146	7.00	146	2.80	146	2.80
			529	12.00	529	4.80	261	7.00	554	2.80	261	2.80
			83	11.00	186	4.50	554	7.00	139	2.40	554	2.80
			491	11.00	594	4.50	669	7.00	218	2.40	669	2.80
			139	10.00	146	2.80	139	6.00	547	2.40	139	2.40
Relevance			85%		95%		92%		93%		92%	
Nodal Reduction							55%		55%			

Table 2: Page rankings for the Computer semantic tree using the six test methods

	Single Word		FT- UW		FT- FW		FT- UW- TC		FT- FW- TC		Comparison	
	Page ID	Score	Page ID	Score	Page ID	Score	Page ID	Score	Page ID	Score	Page ID	Score
Top ranked Pages	46	48.00	157	56.00	46	48.50	46	48.00	46	48.00	46	55.60
	454	48.00	565	56.00	454	48.50	454	48.00	454	48.00	454	55.60
	206	30.00	46	52.00	206	30.10	55	30.00	206	30.00	55	26.80
	614	30.00	454	52.00	614	30.10	206	30.00	614	30.00	463	26.80
	55	28.00	277	40.00	55	28.80	463	30.00	55	28.80	137	25.60
	463	28.00	685	40.00	463	28.80	614	30.00	463	28.80	545	25.60
	137	26.00	156	34.00	137	26.00	135	29.00	137	26.00	206	25.00
	545	26.00	564	34.00	545	26.00	543	29.00	545	26.00	614	25.00
	156	24.00	206	31.00	135	25.90	137	26.00	135	25.90	156	25.00
	564	24.00	614	31.00	543	25.90	545	26.00	543	25.90	564	25.00
	135	23.00	55	30.00	156	25.00	156	24.00	156	24.00	135	23.20
	543	23.00	463	30.00	564	25.00	564	24.00	564	24.00	543	23.20
	73	17.00	135	29.00	210	18.70	210	22.00	210	18.50	210	17.00
	133	17.00	543	29.00	618	18.70	618	22.00	618	18.50	618	17.00
	481	17.00	137	26.00	133	17.60	183	21.00	73	17.50	133	15.40
Relevance	90%		80%		92%		90%		91%		90%	
Nodal Reduction							65%		65%			

Table 3: Page rankings for the ‘Car’ semantic tree using the six test methods

no relevance to the topic/domain in question.

The second worst result in the analysis is the alternate scoring method and pruned tree with unity score which both achieved 92% relevancy. The alternate scoring method has the same fundamental problem as a complete unity tree, as all words are being included in the search. Although hyponym and hypernym scores are not as high as for a unity tree, they are still large enough that if a term is present in a document several times, the page score can be adversely effected.

The best performance of all tests came from an un-pruned, fully weighted word tree which identified 95% of the relevant pages. This result is more consistent with the expected outcomes, unlike the full unity word tree this setup negates the effect of the unproductive words by giving them a low weight in comparison to the root node. Therefore, even if the extended word is mentioned frequently, its weight may be 0.01 and therefore insignificant when compared to other nodes.

The tree threshold function performed better than expected considering the alterations performed to the semantic tree. In all other previous tests (except the single word analysis) the IFA has processed all semantically related words and treated them as being profitable. For the first time the tree pruning algorithm is removing nodes that are unprofitable to the search based on the tree shapes. The impressive part about this result is that the tree pruning algorithm removed over 55% of nodes within the structure and still identified 93% relevant documents identified for the keyword ‘computer’ and 91% relevant document for the keyword ‘car’.

This is evidence that analysing the structure of

a word tree provides meaningful information about the tree’s characteristics, allowing the structure to be altered without degrading the results returned to the user. The analysis identifies nodes within a tree, which are not useful to a search and can return over 96% of the web pages discovered by the best performing function (fully weighted, full tree).

Analysis of tree structures gives rise to improved search results and provides valuable information to a user, but the relevance of the results is not the user’s only concern. If a search engine is too slow for the user due to very high levels of processing being performed, the IFA will not be a viable or acceptable alternative for routine use. Therefore, it is also necessary to examine the overhead and processing time requirements of each method before judging if the tree IFA is successful.

7.4 Hyponym Evaluation

The first impression of hyponym threshold function is that it is rather simplistic. This is due to the nature of most hyponym trees developed by WordNet. There are very few subject words within WordNet, which have any more than a few parent words. Hyponyms on the other hand may consist of many hundred of child words from the specified source. For instance, the word ‘mammal’ has only 8 hyponyms from which it stems but over 1000 different hyponyms. Unless a particularly narrowed and focused word is used as the base of the tree, this generalization rarely changes.

Hyponym trees also have a very particular shapes. Typically they are longer than they are wide as many topics stem from only one or two node. At higher lev-

els the words become very abstract and cover many topics within a much larger domain. This feature provides an advantage when deciding which terms should be cut/pruned from the word tree.

The level of word abstraction plays an important part in the threshold generation function. At the top of all hypernym word trees are terms such as entity or object, which will not usually contribute profitably to a users search. Thus the aim of this function is to gauge how rapidly terms in the hypernym tree become too abstract by examining several different words and their structure. It was concluded that one third of the mean height of the tree was sufficient to include most profitable words. However as stated earlier, a very specific search word may have many useful hypernyms, so one third of them would not be enough. Consequently the function is catered towards low and moderate height hypernym trees whose root word is not very focused towards a particular domain.

$$height = ceil\left[\frac{avg}{\lfloor\frac{avg}{6} + 2\rfloor}\right] \quad (1)$$

To achieve this a linear expression was chosen, as below, using the mean height of all branches as the starting height, then divided by the linear decay.

$$avg = \frac{\sum_{i=0}^{n} branchheight_i}{no.branches} \quad (2)$$

This value was adopted after testing various linear expressions, shown in Table 4 and Figure 7, and their success when applied to several different hypernym word structures. This formula allows for the inclusion of more nodes when analysing larger trees, but has a greater impact on small structures with a fast decay to abstract nodes.

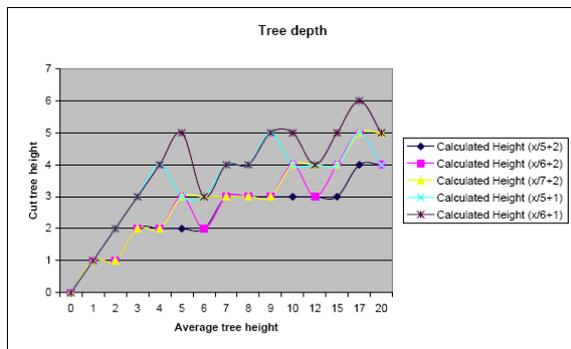


Figure 7: Heights for cutting the hypernym tree (graphical)

As hypernyms rarely exists within a document without associated hyponyms, the analysis of the hypernym function has been incorporated in the hyponym analysis section.

7.5 Execution Time

Given the user applications of IFA's, execution time can be an important factor. If there are 300,000 people requesting a search to be conducted, an increase of 0.099 seconds would have a significant impact on the waiting time and execution of the search. Consequently it is important to examine the impact of the threshold generation and pruning algorithm on the overall search time. The benchmark time for this analysis is the single word search execution outlined in section 7.1. This has been chosen as the benchmark because a typical search engine examines the specified

keyword only and therefore is an accurate approximation of other search engine capabilities. Shown in Table 5 are the average execution times recorded when performing tests on each of the tree structure.

Testing was completed on a Quad processor Intel Xeon server with 2 Gigabytes of RAM and RAID0 SCIS Hard disks. This system was running Linux Redhat Enterprise and Java SDK 1.5.0. This system was chosen for its fast disk access time and available RAM. The execution tests were also conducted on single core desktop computers, a single processor Redhat server and results were not too dissimilar from the Xeon experiments.

The execution time results show a decrease in processing time of up to 10% using a pruned tree compared to the comparison/benchmark function with only 1.1% increase in processing time for the single word analysis. This is considered to be due to the inherent overhead of calculating the word tree thresholds and pruning the word tree structure. All test using derived focus nodes had an increase in execution time, due to the fact that there is more than one word being processed which as expected will take more processing time. Whilst there is a small increase the execution time for the IFA developed in this paper, this is offset by the increase in profitable web pages returned to the user. On the basis that IFA clearly improves the quality and quantity of web search results for users, it is considered that a small increase in processing time can be tolerated.

8 Conclusion

This paper demonstrates that a tree metric IFA can significantly improve searching and filtering of web pages. Evaluation results show a marked improvement using extended word searching as opposed to single word processing. The IFA performed as well as the comparison/benchmark IFA, however due to the reduction of unprofitable related words, there was a very slight decrease in performance and only a slight increase in processing time.

The analysis demonstrates that the threshold generation function's maximum performance was achieved when using a tree with an even distribution of nodes (a tree as wide as it is long), its application on other tree shapes still provided positive results with only a slight decrease in relevance and processing efficiency. Therefore, it is possible to conclude from this study that by analysing underlying semantic tree structures the performance and accuracy of an IFA is improved and more effective than other implementations of extended search IFAs.

References

- Berners-Lee, T. & Conolly, D. (1995), 'Hypertext markup language - 2.0'.
URL: citeseer.ist.psu.edu/article/berners-lee95hypertext.html
- Cesarano, C., d'Acierno, A. & Picariello, A. (2003), 'An intelligent search agent system for semantic information retrieval on the internet', in 'WIDM '03: Proceedings of the 5th ACM international workshop on Web information and data management', ACM Press, New York, NY, USA, pp. 111-117.
- Green, R., Pearl, L., Dorr, B. J. & Resnik, P. (2001), 'Mapping lexical entries in a verbs database to wordnet senses', in 'Meeting of the Association for Computational Linguistics', pp. 244-251.
URL: citeseer.ist.psu.edu/rebecca01mapping.html

Avg. Height	0	1	2	3	4	5	6	7	8	9	10	12	15	17	20
Calculated Height ($x/5+2$)	0	1	1	2	2	2	2	3	3	3	3	3	3	4	4
Calculated Height ($x/6+2$)	0	1	1	2	2	3	2	3	3	3	4	3	4	5	4
Calculated Height ($x/7+2$)	0	1	1	2	2	3	3	3	3	3	4	4	4	5	6
Calculated Height ($x/5+1$)	0	1	2	3	4	3	3	4	4	5	4	4	4	5	4
Calculated Height ($x/6+1$)	0	1	2	3	4	5	3	4	4	5	5	4	5	6	5

Table 4: Heights for cutting the hypernym tree (tabular)

	Single Word	FT-UW	FT-FW	FT-UW-TC	FT-FW-TC	Comparison
ExecutionTime (s)	1.119s	1.143s	1.157s	1.141s	1.121s	1.251
TimeIncrease (%)	0%	+2.1%	+3.3%	+2.1%	1.1%	+9.3%

Table 5: Execution times for conducted experiments

Guandong Xu, Yanchun Zhang, X. Z. (2005), Using probabilistic latent semantic analysis for web page grouping, *in* 'Research Issues in Data Engineering: Stream Data Mining and Applications, 2005. RIDE-SDMA 2005. 15th International Workshop on', pp. 29–36.

Hotho, A., Maedche, A. & Staab, S. (2001), Ontology-based text clustering, *in* 'Proceedings of the IJCAI-2001 Workshop "Text Learning: Beyond Supervision", August, Seattle, USA'.

Lau, R., ter Hofstede, A. H. M. & Bruza, P. D. (2001), Nonmonotonic reasoning for adaptive information filtering, *in* 'ACSC '01: Proceedings of the 24th Australasian conference on Computer science', IEEE Computer Society, Washington, DC, USA, pp. 109–116.

Miller, G., R., B., Fellbaum, C., Gross, D. & Miller, K. (1990), 'Introduction to wordnet: An on-line lexical database', *Journal of Lexicography* **3**(4), 234–244.

Prabowo, R., Jackson, M., Burden, P. & Knoell, H.-D. (2002), Ontology-based automatic classification for the web pages: Design, implementation and evaluation, *in* 'WISE '02: Proceedings of the 3rd International Conference on Web Information Systems Engineering', IEEE Computer Society, Washington, DC, USA, pp. 182–191.

Sim, K. M. (2004), 'Toward an ontology-enhanced information filtering agent', *SIGMOD Rec.* **33**(1), 95–100.

University, S. (2003), 'KSL Ontolingua', <http://www.ksl.stanford.edu/software/ontolingua>.

Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E. G. & Milios, E. E. (2005), Semantic similarity methods in wordnet and their application to information retrieval on the web, *in* 'WIDM '05: Proceedings of the 7th annual ACM international workshop on Web information and data management', ACM Press, New York, NY, USA, pp. 10–16.

Woolridge, M. (2002), *Introduction to Multiagent Systems*, John Wiley & Sons, Inc., New York, NY, USA.

An ontology-based approach for resolving semantic schema conflicts in the extraction and integration of query-based information from heterogeneous web data sources

Abdolreza Hajmoosaei, Sameem Abdul-Kareem

Faculty of Computer Science and Information Technology
University of Malaya
PO Box 50603, Kuala Lumpur, Malaysia

reza_moosaei@yahoo.com, sameem@um.edu.my

Abstract

There are many external resources and heterogeneous data on the internet that an organization or user may need to improve the decision making process. It is therefore, very important and critical that this information are complete, precise and can be acquired on time. Most web sources provide data in semi-structured form on the internet. The combination of semi-structured data from different sources on the internet often fails because of syntactic and semantic differences. The access, retrieval and utilization of information from the different web data sources impose a need for the data to be integrated. Integration of web data is a complex process because of the heterogeneity nature of web data and thus needs some kind of a web data integration system. There are many types of heterogeneity and differences among web sources that makes data integration a difficult process (e.g., different data model, different syntax and semantics in schema and data instance level among web sources). Semantic schema heterogeneity, which refers to the misinterpretation of data at the schema level, is one major obstacle that needs to be overcome in web data integration process. Semantic schema heterogeneity has been identified as one of the most important problems when dealing with interoperability and cooperation among multiple data sources on the internet. In this paper, we recommend a system architecture for web data integration focusing on resolving the problems of semantic schema heterogeneity between web data sources. We propose an ontology-based approach as a solution for the reconciliation of semantic conflicts between web data at the schema level.

Keywords: Web data integration, Semantic schema heterogeneity, Ontology.

1 Introduction

The web is the platform for information publishing; it is the biggest resource of information of any type. There are a lot of valuable data and business data on the web that organizations or users can use to improve their decision

Copyright(c) 2007, Australian Computer Society, Inc. This paper appeared at the 3rd Australasian Ontology Workshop (AOW-07), Gold Coast, Queensland, Australia. Conferences in Research and Practice in Information Technology, Vol. 85. Editors, Thomas Meyer and Abhaya C. Nayak. Reproduction for academic, not-for profit purposes permitted provided this text is included.

making process. It is therefore, very important and critical that this information is complete, precise and can be acquired on time (Heflin and Hendler 2000). It is also vital that such external information be systematically managed and utilized for users. Each information system on the web is modelled and implemented differently according to the requirements of the application domain. The access, retrieval and utilization of information from the different data sources imposes a need for the data to be integrated because there are many types of heterogeneity and differences among web sources that makes a combined effort to access data from different sources on the internet difficult and error-prone (Kashyap and Sheth 1998) (Fensel *et al.* 1999). The following HTML pages from different room reservation systems as shown in Figures 1 & 2 illustrate this. For example, if a user queries rate of hotel rooms, the retrieval and combination of data related to rate of rooms in the following HTML pages fail because they have heterogeneity conflicts to each other such as: use different names (“price” and “Rate”), different units (“EUR” and “USD”) to represent cost of rooms.

Room reservation, KL, Malaysia on 07/04/2005			
Select	Hotel	type	Price
	Gurney	single	EUR 57.99/day
	Rose	double	EUR 75.99/day

Fig. 1. Reservation System A

Room booking-Kuala Lumpur		Apr. 07 2005
Rate :	Daily Rate	USD 67.99
Hotel :	Gurney	
Room type :	Standard/single	
SELECT		

Fig. 2. Reservation System B

The solution to the problem mentioned above is a web data integration system. External information can be extracted from web sources and utilized for users through a web data integration system. The design of such a system is not easy because the differences in web data make data integration a difficult process. Integration of heterogeneous data sources from the internet is a complex activity that involves reconciliation of various levels of

conflicts. Before we can integrate the heterogeneous data we need to resolve these heterogeneity conflicts. There are different views about classification of Heterogeneity conflicts. The heterogeneity conflicts can be classified according to the following abstraction levels (Ram and Park 2004) (Kashyap and Sheth 1998):

- *Data Value Conflicts*: Data value conflicts are those conflicts that arise at the instance level. They are related to the representation or the interpretation of the data values. Examples of these conflicts are discrepancies of type, unit, precision and allowed values (e.g. "kg" and "gram" or "\$" and "dollar").
- *Schema Conflicts*: Schema conflicts are due to different alternatives provided by one data model to develop schemas for the same reality. For example, what is modelled as an attribute in one relational schema may be modelled as an entity in another relational schema for the same application domain (e.g. "Author" as attribute for the entity "book" and "author" as an entity that has a relationship with "book"). Another example two sources may use different names to represent the same concept (e.g. "price" and "cost") , or the same name to represent different concepts , or two different ways, for conveying the same information(e.g. "data of birth" and "age").
- *Data Model Conflicts*: Data model conflicts occur when databases use different data models, e.g., one database designed according to the relational model, and another one object-oriented.

Conflicts in each level can be categorized into two categories:

- *Syntactic Conflicts*: Syntactic conflicts refer to discrepancies in the representation of data (e.g. "1/23" and "1.23" or "price=23\$" and "price: 23\$").
- *Semantic Conflicts*: Semantic conflicts refer to disagreement about the meaning, interpretation use of the same or related data (e.g. "staff" and "employee").

The major aim of our work is to give a solution for resolving semantic schema heterogeneities in a web data integration system. For this purpose we first recommend a system architecture for web data integration and subsequently propose an approach to resolve semantic conflicts in this system. We use ontology as a solution for the reconciliation of semantic conflicts between web data at the schema level.

The ontology is one effective solution to semantic heterogeneity problem in web data integration. Explicit semantic information of terms in the ontology can help in resolving semantic heterogeneities among web data (Fensel 2001). The subject of ontology is the study of the categories of things that exist or may exist in some domain. The product of such a study, called ontology, is a catalogue of the types of things that are assumed to exist in a domain (Sowa 2000). Ontology helps to figure out what a specific term means. Ontologies provide a way to

describe the meaning and relationships of terms so that a shared understanding or a consensus can be acquired among people and machines (Guarino 1998). The reason ontologies are becoming so popular is in large part due to what they promise: a shared and common understanding of some domain that can be communicated between people and application systems (Fensel 2001).

In our approach, we assume that each web source have its own ontology. Therefore any web source is free to have its own vocabulary and semantic independence from other web sources' vocabulary. Independency of web sources in defining their ontologies may cause a major problem. We need the consensus of the communities (web sources) over the meaning of the terms in order to resolve any problems that may arise from semantic conflicts. This consensus in the web data context is not feasible because communities are free to use their own vocabularies and semantics according to their requirements. Therefore any web source can have its own ontology containing all the defined terms that has met with the agreement of a group of users. This problem can be resolved through semantic mapping between ontologies (Hakimpour and Geppert 2001). In the semantic mapping process, a reasoning system finds the similarities concepts between two ontologies and maps the corresponding concepts to each other. Semantic ontology mapping is one of the main tasks of a web data integration process. In this paper we propose an approach to semantically map ontologies and use this mapping in the integration process of our system.

2 Related work

There are many proposed approaches and systems for semantic data integration by researchers. In this section we chose five major projects that have been the foundation for other projects and researches. These projects focus on the use of ontologies for resolving semantic conflicts.

SIMS is a system that extracts a semantic data model of an application domain to integrate the information from various information sources. This semantic data model includes a hierarchical terminological knowledge and has the role of a global ontology in SIMS. SIMS uses a data model from each information source and these data models play the role of local ontologies. Each data model of information resource must be described for this system by relating the objects of the data model to the global domain model. The relationships clarify the semantics of the source objects and help to find semantically corresponding objects. In SIMS the user formulates a query in terms of the global domain model. Then SIMS reformulates the global query into sub-queries for each appropriate source, collects and combines the query results, and returns the results (Arens, Ciiee and Knoblock 1992).

The COIN project presents an architecture for semantic interoperability between distributed information sources. The COIN framework uses a data model and logical language to define the domain model of the application and the contexts. The domain model plays the role of the

ontology in the COIN-framework. Context mediation in Coin-architecture performs the process of rewriting queries posed in the receiver's context into a set of mediated queries where all potential conflicts are explicitly resolved. This process is according to the statements in the different contexts involved, what information is needed to answer the query and what and how conflicts may be resolved (Goh, *et al.* 1999).

MOMIS (Mediator Environment for Multiple Information Sources) is one approach to the integration and query of semi-structured and structured heterogeneous data sources (Beneventano, *et al.* 2001). The goal of MOMIS is to define a global schema that allows uniform and transparent access to the data stored in a set of semantically heterogeneous sources. MOMIS creates a global virtual view (GVV) of information sources, independent of their location or their data's heterogeneity. MOMIS builds ontology through five phases as follows: 1) Local source schema extraction by wrappers; 2) Local source annotation with the WordNet; 3) Common thesaurus generation: relationships of inter-schema and intra-schema knowledge about classes and attributes of the source schemas. 4) GVV generation: A global schema and mappings between the global attributes of the global schema and source schema by using the common thesaurus and the local schemas are generated. 5) GVV annotation is generated by exploiting annotated local schemas and mappings between local schemas and a global schema.

The *KRAFT* architecture is designed to support *knowledge fusion* from distributed, heterogeneous databases and knowledge bases (Visser, *et al.* 1999). KRAFT is a project for the integration of heterogeneous information, using ontologies to resolve semantics problems. They extract the vocabulary of the community and the definition of terms from documents existing in an application domain. KRAFT detects a set of ontology mismatches and establishes mappings between the shared ontology and local ontologies.

OBSERVER is an approach for query processing in global information systems based on interoperability across pre-existing ontologies (Mena, *et al.* 1996). OBSERVER allows users to pose their queries by using ontologies against heterogeneous data sources. It replaces terms in user queries with corresponding terms in data source ontologies by detecting similarity relations between user ontology and local ontology. OBSERVER uses Description Logic as both ontology definition language and query language. The OBSERVER is the foundation of our proposed web data integration system.

For reconciliation of semantic conflicts between heterogeneous data sources, the above mentioned projects create one global or shared ontology by integrating or merging local schemas or ontologies. Subsequently, they perform semantic mapping between created global ontology and the local schemas or ontologies. In the web context the maintenance and updating of global or shared ontology is very time consuming and costly because many web data sources are involved and the number of involved web data sources change frequently; web designers and users are free to use their own terms and

vocabulary and schemata which are subject to frequent changes. In our proposed approach we try to overcome this problem by using domain specific ontology and we resolve semantic problems between web sources through semantic mapping between the domain ontology and the local ontologies.

3 System architecture

This system (Figure 3) uses ontologies for resolving semantic schema conflicts between web data sources. The system uses domain specific ontologies for the creation of user queries. There is a domain specific ontology for each application domain that covers the semantic definition of terms which are required for user query in a particular application domain. The domain ontologies are modelled in an internal uniform representation model. The user can browse the domain ontology and choose terms for his/her query, the system then creates the user query. We assume each web source has an underlying pre-existing local ontology on the web and each local ontology is associated with one or more web sources. After the creation of the user query, the web ontology server chooses local ontologies related and relevant to user query domain and sends them to the mapping module. The local ontologies are transformed to the internal uniform representation model by transformers. Afterwards, the user query terms are mapped to corresponding terms in the local ontology and the user query is subsequently rewritten using the terms from the local ontology.

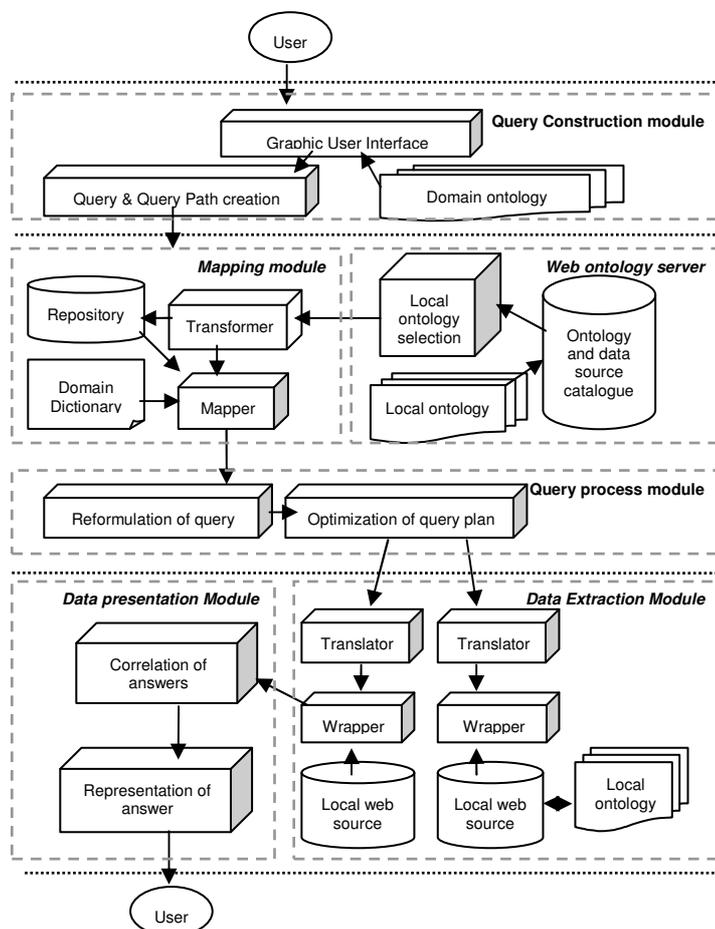


Fig. 3: Semantic web data integration system

The rewritten user query is sent to a query process module for reformulation and creation of optimised query plan from the user query. Finally, the gained sub queries from reformulation process are translated to the web sources query languages by translators and the data is extracted by wrappers and presented to the user.

The proposed system resolves semantic schema problems between web sources and user query through semantic mapping between the domain ontology and local ontologies. We use inter-mappings between domain and local ontologies for semantic integration of user query terms and web data sources terms.

4 Semantic Integration Process

The proposed web data integration system implements a query based approach to information extraction and integration, from heterogeneous and distributed web data sources. The system possesses a three layer architecture as follows:

- Data and physical layer in this system consists of a data extraction module,
- Application layer consists of a query process module, a mapping module and web ontology servers,
- Presentation layer consists of a query construction module and a data presentation module.

The extraction and integration process in proposed system consists of eight major tasks as follows:

1. Creation of user query and query path;
2. Determination of related local ontologies with query domain;
3. Transformation of related local ontologies to internal uniform representation model;
4. Semantic mapping between query terms and related local ontologies terms;
5. Rewriting of user query with corresponding terms from local ontologies;
6. Reformulation of query and creation of optimized query plan;
7. Translation of sub queries to web sources query languages and extraction of answers
8. Correlation and representation of answer.

In the rest of this paper we focus on semantic mapping and query construction modules of the proposed system and suggest our approach for resolving semantic schema conflicts between user query terms (chosen from domain ontology) and related local ontologies terms. Our approach covers the first, fourth and fifth tasks of the integration process mentioned above.

4.1 Domain and Local Ontologies

Our proposed system uses ontologies as a solution for reconciliation of semantic schema heterogeneities

between web data sources. The system exploits two types of ontologies: domain specific ontology and local ontology. Domain ontology is created for one specific domain. For example if the system has been developed for the domain of a university, so one university specific ontology is designed and created for the system. The users of the system choose their query-terms from the domain ontology and they are not free to use their preferred terms. User query terms are restricted to the domain ontology terms.

There are many ontologies on the web that are used for the semantic description of data. We call these ontologies as local ontologies. In our system we assume any web source has an underlying local ontology. Therefore, in order to resolve semantic conflicts between user query terms (chosen from the domain ontology) and the terms from web resources, a semantic mapping between the user query terms and the related local ontologies terms is needed. This semantic mapping relates similar terms from the two different sources by specifying the correspondence between them.

4.2 Uniform Representation of Ontology

The mapping module of the system finds similar and corresponding terms between the related local ontology and domain ontology and maps them to each other. Local ontologies on the web have been formalized in different models and languages. In order to compare and find similar terms between the domain and local ontology, the system needs to represent all ontologies in a uniform model. In our system we propose one uniform representation model for ontologies. This representation model is general and any ontology with any representation model can be transformed to this uniform representation model.

Definition1: $T:=(C,A,R,V)$, each ontology element (term) is one of following entities:

- C: concept or instance of one concept
- A: attribute of one concept
- R: relationship between concepts
- V: value range of one relationship

For example student (concept), age (attribute), master student (instance of student, it is considered as sub-concept of student in our model), attend (relationship between student and class) and “<20” (value range of “max-credit-course” relationship) are some element (term) of university ontology.

Definition2: $C:=(name, syn-set, A, key-A, key-R)$, each concept is defined with its name, set of its synonyms, attributes, its key attributes, and key relationships with other concepts. The key attributes are subset of concept attributes. The key attributes and key relationships are specific properties and specifications of one concept that characterize the concept. These key properties are specified just for concept definitions of the domain ontology during the development of the domain ontology. We will use these properties as a mapping criterion for finding similar terms in our mapping algorithm.

Definition 3: $A := (name, syn-set)$, attribute is defined with a name and a set of synonyms.

Definition 4: $R := (name, syn-set, domain, range)$, each relationship is defined with a name, set of synonyms and domain and range.

Definition 5: $V := (value)$, this feature is used for representation range of one relationship that is a value. One value Begins with one of these characters: "=", "<", ">" or "< >" and one string that show the value of its range.

Definition 6: $O := (G, G')$, each ontology is represented by two graphs.

Definition 7: $G := (N, E)$, $N = \langle C \rangle$, $E = \langle is-a \rangle$, G is acyclic directed rooted graph that consists of nodes and edges. Each node is a concept (or instance of a concept). Each edge is "is-a" relation that shows sub-concept (subclass) relation between nodes. Indeed, G is a hierarchy concept model of ontology. Each node has one father and may have no, one or more child nodes. If one node has two fathers, the model resolves this problem with repeating child node for each one of its fathers.

Definition 8: $G' := (N, E')$, $N = \langle C, V \rangle$, $E' = \langle R \rangle$, G' is cyclic graph that consists nodes and edges. Each node is a concept (or instance of a concept) or one value. Each edge is relationship between two nodes that show the relationship between concepts. Indeed, G' is a concept relationship model of ontology.

In a uniform representation model, all elements (concepts, attributes, relationships and values) are string (chain of characters). Our representation model and formalization of ontology is very general, so our proposed approach which uses this formalization will work with any ontology representation languages. We need to transform (in mapping module of system) the local ontology to the uniform representation model. This representation model represents the main exploitable information in an ontology and by exploiting all of the available information which we have, the calculation of the similar concepts and semantic mappings between the domain ontology (query terms) and the local ontology will gain a better result in quality.

We use the table structure (Relational Database) to store ontologies represented in the uniform representation model using any DBMS implementation such as MySQL.

4.3 User Query construction

Our proposed system is a domain specific system because the user is confined to choosing terms of a specific domain ontology for his/her query. This system can be extended for any domain so that the relevant domain ontology would be developed previously in the system. The user is confined to use just one domain ontology for his/her query. Users can not pose complex queries because the query construct and structure in this system is based on the structure and elements of the internal uniform representation model of the domain ontology. We define the following structure and syntax for the expression of the user query.

```
SELECT < attributes names >
FROM < concept name >
WHERE
{
<attribute names: values> FROM <concept name1>
AND/OR
<attribute names: values> FROM <concept name2>
.....}
```

In this query structure, the user can query the attributes of only one concept from the domain ontology. The user can specify constraints and conditions on his/her query. Constraints and conditions are expressed after the "WHERE" clause in the query expression.

We clarify the query syntax and structure with the following example:

Suppose that a user needs the names and emails of professors in Law at universities who are above 55 or below 35 years of age and are female. This query is in the university domain, so we assume there is a university specific ontology in the system. So the user traverses terms in the university ontology and chooses his/her query terms. Afterwards, the system constructs an expression based on the user query as follows:

```
SELECT name, Email
FROM Professor
WHERE
{
name =Law FROM Department AND
sex =Female FROM Professor AND
age >50 FROM Professor OR
age <35 FROM Professor
}
```

Note that the user can not pose complex queries. The user can split his/her complex query to simple sub-queries and subsequently submit them to the system. For example, if a user has a query about attributes relating to two concepts, then he/she must pose two separate queries to the system. This is applied to the constraint and condition segments in the query as well. The user can simply specify conditions for the attributes of concepts which are in the query path. The path through which a user traverses in the domain ontology graphs for reaching his/her query terms is called query path. For example in the above query, conditions which have been specified for the attributes of name, sex and age all are related to the concepts in the query path. We discuss more on query paths in the next section.

4.4 Query Path

In query construction process the user traverses the internal uniform representation graphs of the domain ontology in order to choose his/her query terms. The path through which a user traverses in the domain ontology graphs for reaching his/her query terms is called query path. We use this query path as comparison and mapping criteria for finding similar and corresponding terms between the user query and the local ontologies in the semantic mapping algorithm.

A user interacts with the system through GUI (user graphic interface) in the query construction module of the system. This GUI must display the domain ontology to the user so that user can find his/her query terms easily and quickly. In the time the user takes to interact with the system for the construction of the query, the system performs two tasks: first finding of query terms and creation of query and second specifying of query path for use in the semantic mapping algorithm.

In the meantime, the system obtains information from the user for the creation of the query path. How we design the GUI and display the graph models of domain ontology to the user, have a role in precise and exact creation of the query path. In our approach, the GUI first displays a list of domain ontologies to the user. The user chooses one domain ontology related to his/her query. Then, the GUI displays super concepts of domain ontology to user (concepts in top of hierarchy concept graph). The user chooses one of the super concepts. This chosen concept is the root of the query path. We call it T_1 . In next step the GUI shows three types of information to the user related to T_1 that are choices for the user for a second term of query path (we call T_2):

- Sub-concepts (children) of T_1 ,
- Attributes of T_1 ,
- Relationships and their ranges which T_1 is the domain of those relationships.

Figure 4 provides a partial illustration of this information about the concept of *student*.

<p>Sub concepts of Student:</p> <ul style="list-style-type: none"> • Undergraduate student • Postgraduate student • Post doctoral student • Non-graduating student • Exchanging student 	<p>Attributes of student:</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;"><input type="checkbox"/> Name</td> <td style="width: 15%;">and/or</td> <td style="width: 15%;"><input type="text"/></td> <td style="width: 15%;">Value</td> <td style="width: 15%;"></td> <td style="width: 15%;">query</td> </tr> <tr> <td><input type="checkbox"/> Age</td> <td>and/or</td> <td><input type="text"/></td> <td></td> <td></td> <td>Y/N</td> </tr> <tr> <td>.....</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table> <p style="text-align: center; margin-top: 10px;"><input type="button" value="SUBMIT"/></p> <table style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <tr> <td style="width: 50%;">Relationship</td> <td style="width: 50%;">Range</td> </tr> <tr> <td style="border: 1px solid black; border-radius: 10px; padding: 5px; text-align: center;">Advisor-by</td> <td style="text-align: center;">→ Faculty</td> </tr> <tr> <td style="border: 1px solid black; border-radius: 10px; padding: 5px; text-align: center;">Study-in</td> <td style="text-align: center;">→ Program</td> </tr> </table>	<input type="checkbox"/> Name	and/or	<input type="text"/>	Value		query	<input type="checkbox"/> Age	and/or	<input type="text"/>			Y/N						Relationship	Range	Advisor-by	→ Faculty	Study-in	→ Program
<input type="checkbox"/> Name	and/or	<input type="text"/>	Value		query																				
<input type="checkbox"/> Age	and/or	<input type="text"/>			Y/N																				
.....																									
Relationship	Range																								
Advisor-by	→ Faculty																								
Study-in	→ Program																								

Fig. 4: Example of GUI of university ontology

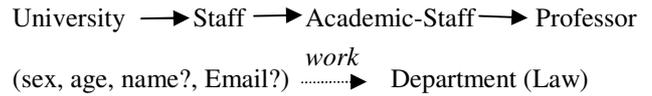
In order to choose the next term, the user has the following choices:

If a query has a condition or a constraint on the attributes of T_1 , then user enters the constraint value of the attributes in the value fields and determines “and/or” relate to each attribute value (constraint) with the previous attribute value. After determination of constraint attribute values, user follows one of these steps :

- Choose one of sub-concepts; or

- Choose one of ranges (if range is concept element, no value element); or
- If this concept is a query concept in which a user needs to query its attributes then: choose “Y” for query attributes.

In this way the query construction module obtains the query path and finally creates the user query. The Query path contains all the terms that the user traverses in mentioned above steps. The query path does not consist any values for the attributes but just the attribute names. For example query path for mentioned query in section 4.3 from university ontology can be:



We consider following definition for query path:

Definition 9: $QP := (N_p, E_p, E'_p)$, a acyclic directed rooted path from uniform representation graphs (G & G') of domain ontology that traversed by user.

Definition 10: $N_p := (C\text{-name}, A_1\text{-name}, A_2\text{-name}...)$, is a concept or instance name (C) and its attached attributes that these attributes specified by user as constraints or query attributes on this concept in query construction time.

Definition 11: $E_p := (is-a)$, it shows sub-concept relationship between concepts in query path.

Definition 12: $E'_p := (R\text{-name})$, it represents relationship name.

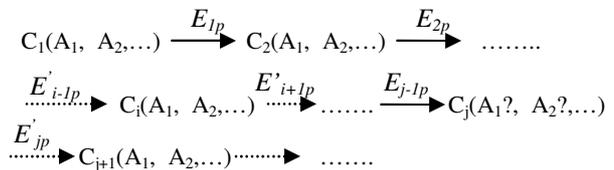


Fig. 5: Structure of Query Path

4.5 Semantic mapping

After creation user query and query path by query construction module, query needs to be translated to relevant web sources query languages. For this translation the first, user query terms must be semantically mapped to similar terms in local ontologies of related web sources. This semantic mapping relates each query terms with its semantically similar and corresponding term of related local ontology. Indeed, the semantic mapping is performed between partial of domain ontology (related to query path terms) and local ontology. Therefore we need to algorithm for semantic mapping between domain and local ontologies.

There are approaches for semantic mapping between ontologies that have been proposed by researchers. Some of recent researches and approaches in ontology mapping domain are Chimaera (McGuinness *et al.* 2000), Anchor-PROMPT (Noy and Musen 2001), QOM (Ehrig and Staab 2001), Cupid (Madhavan *et al.* 2001), GLUE

(Doan *et al.* 2002), SAT (Giunchiglia and Shvaiko 2003) and ASCO (Thanh-Le *et al.* 2004). Our approach was motivated by some ideas of the above approaches. Above approaches exploit available information from ontologies and map similar terms of two given ontologies to each other by mapping algorithms.

We can evaluate an ontology mapping approach base on two main factors: quality of mapping results and run time complexity of mapping approach. Some approaches have high quality mapping result but they are not applicable because of high runtime of their mapping algorithm. Therefore we must consider both factors in design of mapping algorithm. We have no space in this paper for discussion about above approaches and problems of ontology mapping. For this purpose we refer reader to (Klein 2001).

4.5.1 Semantic mapping algorithm

Inputs of our mapping algorithm are: query path, domain ontology and local ontology. There are three types of ontology element in query path: concept (C), attribute (A) and relationship (R). All elements are string (chain of characters) and may be a word, term or expression (combination of words). The query path consists user query terms and some un-query terms. The un-query terms are terms which have been traversed by user for reaching to query terms.

The purpose of mapping algorithm is finding of semantically similar terms with query terms from local ontology and then rewriting of query with finding similar terms. For calculation of similarity between two elements from query path and local ontology, following function is used in mapping algorithm:

MF (Mapping Function): $MF(element_1, element_2) := [0, 1]$; This function calculates similarity between two elements. Value rang of $[0, 1]$ indicates amount of similarity. MF performs two sub-functions for similarity calculation. First sub-function, normalizes two elements to their tokens. In this sub-function each term (concept, attribute or relationship) be:

- Tokenized: <Hands-free> <Hands, Free>
- Lemmatized: <Kits> <Kit>
- Eliminated: <courtforplay> <Court, Play>

For normalization, it exploits and uses one domain specific dictionary. This dictionary consists all existing terms in a specific domain include their synonym sets.

The second sub-function, compares tokens (string without space) of normalized terms with each other and calculates similarity between tokens. Finally, similarity between two elements is calculated from aggregation of token similarities. There are well-known metrics for calculating string similarity between two tokens such as Jaro-Winkler metric (JW), Levenstein metric and Monger-Elkan (Cohen *et al.* 2003). We can apply one of them in our mapping algorithm.

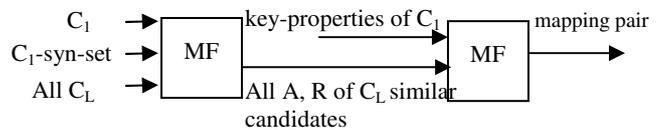
The main Steps of our mapping algorithm are as follows (consider structure of query path in figure5):

First step: MF is executed between $C_I(name)$ (root of query path), all its synonyms names $C_I(syn-name)$ with all local ontology concepts (all $C_L(name)$).

```
for all  $C_L(name) \neq null$  do
  { If  $MF(C_I(name), C_{iL}(name)) \geq$  threshold then
    Add ( $C_I, C_{iL}$ ) to similarity-table;
  Else: for all  $C_I(syn-name) \neq null$  do
    { If  $MF(C_I(syn-name), C_{iL}(name)) \geq$  threshold
      then Add ( $C_I, C_{iL}$ ) to similarity-table; }
```

Result of above step is some similar pairs, which have similarity measure above algorithm threshold. We call them, candidate mapping pairs. Now algorithm must find best similar pair among candidate mapping pairs. So, algorithm executes MF between key attributes and key relationships of C_I with all attributes and relationships of its similar elements in each candidate mapping pair. We are going to find best and closest matching and similar pair. We choose maximum MF that is above threshold. This similar pair $\langle C_I, C_{iL} \rangle$ is stored in *C-mapping-table* as final result of MF for C_I . C_{iL} is mapping element of C_I from local ontology.

```
while  $C_{iL}(name) \neq null$  do
  { While  $C_{iL}(A) \neq null$  or  $C_{iL}(R) \neq null$  do
    If each  $MF(C_I(key-property-name \& key-property-syn-names), C_{iL}(A \& R)) \geq$ 
      threshold then  $C_{iL}(similar-property) + 1$ ;
     $C_{iL} [MAX C_{iL}(similar-property)]$ ;
    Add ( $C_I, C_{iL}$ ) to C-mapping-table;
```



Second step: After finding similar concept of C_I , If C_I has attribute in query path then we must find its similar attributes in local ontology. We should notice, we just execute MF between C_I -attribute-name, all C_I -att-synset-names with attributes-names and relationships-names of its mapping pair (C_{iL} in mapping table). We choose maximum MF that is above threshold and store similar-attribute pairs in *C-att-mapping table* (such as: $\langle C_I-A_1, C_{iL}-A_{iL} \rangle, \langle C_I-A_2, C_{iL}-A_{2L} \rangle, \dots$).

```
while  $C_I(A-name) \neq null$  do
  { If  $MF(C_I(A-name), C_{iL}(A-name \text{ or } R-name)) \geq$ 
    threshold then
    Add ( $C_I(A-name), C_{iL}(A-name \text{ or } R-name)$ ) to att-
    mapping-table;
  Else: while  $A-syn-name \neq null$  do
    { If  $MF(C_I(A-syn-name), C_{iL}(A-name \text{ or } R-name)) \geq$ 
      threshold then
      Add ( $C_I(A-name), C_{iL}(A-name \text{ or } R-name)$ ) to
      att-mapping-table; }
```

Third step: We must find similar concept for next term of query path (C_2). There are two situations here: C_2 has “is-a” relationship with C_1 (C_2 is sub-concept of C_1) or C_2 has “R” relationship with C_1 (C_1 and C_2 are domain and range of same R).

In first situation, algorithm follows tasks of **first step** for finding similar concept of C_2 just with this difference that algorithm doesn't compare (MF) C_2 with all of local ontology concepts. As C_2 is sub-concept of C_1 , then:

- Algorithm first compares (MF) C_2 with children of C_{1L} .
- If algorithm could not find similar concept of C_2 in children of C_{1L} then it compare (MF) C_2 with siblings of C_{1L} .
- If algorithm could not find similar concept of C_2 in siblings of C_{1L} then it compares (MF) C_2 with all concepts that are range of relationships which C_{1L} is domain of those relationships.
- Finally if again algorithm could not find similar concept of C_2 then it compares (MF) C_2 with C_{1L} . Because C_{1L} may be further general and cover semantics of both of C_1 and C_2 .

If algorithm found similar concept of C_2 then stores similar pair $\langle C_2, C_{2L} \rangle$ in *C-mapping-table* and then executes MF for query path attributes of C_2 (**second step** of algorithm). If algorithm doesn't find similar concept of C_2 then store pair $\langle C_2, null \rangle$ in *C-mapping-table* and if C_2 has attribute in query path then algorithm enters $\langle C_2-A_1, null \rangle, \langle C_2-A_2, null \rangle \dots$ in *att-mapping-table*. In this case algorithm uses C_{1L} instead of C_{2L} in next steps of algorithm, because C_{2L} is null.

In second situation (C_2 has "R" relationship with C_1), algorithm performs following tasks:

Algorithm executes MF between R , R -synonyms with all relationships of C_{1L} (C_{1L} is domain of relationships).

- If algorithm found similar relationship then:
 - Executes MF between C_2 and ranges of discovered similar relationship. If it finds similar concept of C_2 then enters similar pair $\langle C_2, C_{2L} \rangle$ in *C-mapping-table* and then executes MF for query path attributes of C_2 (**second step** of algorithm)
 - else enters $\langle C_2, null \rangle$ in *C-mapping-table*.
- If algorithm doesn't find similar relationship with R then it executes tasks of **first step** of algorithm for C_2 . That means algorithm executes MF between C_2 and all of local ontology concepts. If it finds similar concept of C_2 then enters similar pair $\langle C_2, C_{2L} \rangle$ in *C-mapping-table* and executes MF for query path attributes of C_2 (**second step** of algorithm) else enters $\langle C_2, null \rangle$ in *C-mapping-table*.

Algorithm repeats **third step** for next others nodes until last element of query path. If algorithm doesn't find similar concept of main query concept (concept in question) from local ontology so, mapping doesn't execute between user query terms and local ontology terms and this local ontology is failed.

Specifications of algorithm

- The specifications of our semantic mapping algorithm are as follows:
- It is in element level no structure level. Because it finds similarity in element granularity (concept, attribute and relationship similarity).
- It uses $C(name)$, $C(syn-set)$, $C(key-A)$ and $C(key-R)$ as mapping criteria for concepts.
- It uses $R(name)$ and $R(syn-set)$ as mapping criteria for relations.
- It uses $A(name)$ and $A(syn-set)$ as mapping criteria for attributes.
- It is linguistic-based because it finds similarities by element-name matching and element-synonym matching (string matching).
- It is also constraint-based because it uses key properties (key-A and key-R) of concept for finding best similarity.
- It is path-based because it uses path of query. It considers semantic relations in path and restricts domain of search in local ontology.
- It has 1:1 mapping local cardinality. Because it maps each query term to one local ontology term no more.
- It uses domain specific dictionary as auxiliary information for help to finding similarities.
- It can be applied for any ontology model or language because it uses one general representation model for ontologies.
- Runtime complexity of algorithm is: $m \cdot n$ $O(MF)$.
- m is maximum length of query path, n is number of all local ontology terms (in worst situation) and $O(MF)$ is run time complexity of matching function.
- Output of algorithm are c-mapping-table (concept mappings) and att-mapping-table (attribute mappings).

5 Conclusion

In this paper, we first recommended a system architecture for semantic schema web data integration and subsequently proposed an approach to resolve semantic schema conflicts in this system. We use ontology as a solution for the reconciliation of semantic conflicts between web data sources at the schema level. We introduced one uniform graph-based representation model for ontologies and proposed approach for creation of user query base on this representation model. We proposed one semantic mapping algorithm. Our mapping algorithm exploits user query path for finding similarities between user query terms and local ontology terms.

6 Reference

- Heflin, J., and Hendler, J., 2000: Semantic interoperability on the web. In *Extreme Markup Languages 2000*. <http://www.cs.umd.edu/projects/plus/SH/OE/pubs/extreme2000.pdf>.
- Kashyap, V., and Sheth, A. 1998: Semantic heterogeneity in global information systems: The role of metadata, context and ontologies. In *Papazoglou, M. P. and Schlageter, G., editors, Cooperative Information Systems: Current Trends and Directions, pages 139–178. Academic Press Ltd.*
- Fensel, D., Angele, J., Decker, S., Erdmann, M., Schnurr, H., Staab, S., Studer, R. and Witt, A. 1999: On2Broker: Semantic-based access to information sources at the WWW. In *Proceedings of the World Conference on the WWW and Internet (WebNet 99), pages 366–371.*
- Ram S., Park, J. 2004: Semantic Conflict Resolution Ontology (SCROL): An Ontology for Detecting and Resolving Data and Schema-Level Semantic Conflicts, *IEEE Transactions on Knowledge and Data Engineering, v.16 n.2, p.189-202.*
- Fensel, D. 2001: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. *Springer-Verlag.*
- Sowa, J. F. 2000: Guided tour of ontology <http://www.jfsowa.com/ontology/guided.htm>.
- N. Guarino, editor (1998). *Formal Ontology in Information Systems*. IOS Press, Amsterdam.
- Hakimpour, F. and Geppert, A. 2001: Resolving semantic heterogeneity in schema integration: An ontology base approach. In *Welty, C. and Smith, B., editors, Formal Ontology in Information Systems: Collected Papers from the Second Int'l Conf., FOIS'01, pages 297–308. ACM Press.*
- Arens, Y., Ciiee, Y., Knoblock, A. 1992: SIMS: Integrating data from multiple information sources. *Information science institute, University of Southern California, U.S.A.*
- Goh, C.H., Bressan, S., Madnick, S. and Siegel, M. 1999: Context interchange New features and formalisms for the intelligent integration of information. *ACM Transaction on Information Systems, 17(3):270–290.*
- Beneventano, D., Bergamaschi, S., Guerra, F. and Vincini, M. 2001: The MOMIS approach to information integration. In *ICEIS 2001, Proceedings of the 3rd International Conference on Enterprise Information Systems, Portugal.*
- Visser, P. R., Jones, D. M., Beer, M., Bench-Capon, T., Diaz, B. and Shave, M. 1999: Resolving ontological heterogeneity in the KRAFT project. In *10th International Conference and Workshop on Database and Expert Systems Applications DEXA'99. University of Florence, Italy.*
- Mena, E., Kashyap, V., sheth, A. and Illarramendi, A. 1996: OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies.
- MySQL: SQL Shareware Software, MySQL AB Co. <http://www.mysql.com/>. Accessed 29 Dec 2001.
- McGuinness, D.L., Fikes, R., Rice, J., and Wilder, S. 2000: The chimaera ontology environment, in *Seventh National Conference on Artificial Intelligence (AAAI-2000).*
- Noy, N. F. and Musen, M. A. 2001: Anchor-PROMPT: Using Non-Local Context for Semantic Matching. *Workshop on Ontologies and Information Sharing. IJCAI, Seattle, WA, 2001.*
- Ehrig, M. and Staab, S. 2001: Efficiency of Ontology Mapping Approaches, *Institute AIFB, University of Karlsruhe.*
- Madhavan, J., Bernstein, P. A., and Rahm, E. 2001: Generic Schema Matching with Cupid. In *Proc. Of the 27th Conference on Very Large Databases.*
- Doan, A., Madhavan, J., Domingos, P. and Halevy, A. 2002: Learning to Map between Ontologies on the Semantic Web. *The Eleventh International World Wide Web Conference (WWW 2002), Hawaii, USA.*
- Giunchiglia, F. and Shvaiko P. 2003: Semantic Matching. *CEUR-WS, vol: 71.*
- Thanh Le, B., Dieng-Kuntz R., Gandon, F. 2004: On Ontology Matching Problems: for building a corporate Semantic Web in a multi-communities organization, *Institute National and Research in Informatic, Sophia Antipolis, France.*
- Klein, M. 2001: Combining and relating ontologies: an analysis of problems and solutions, *Vrije University Amsterdam.*
- Cohen, W., Ravikumar, P. and Fienberg, S. 2003: A Comparison of String Distance Metrics for Name Matching Tasks. *IJCAI 2003, workshop on Information Integration on the Web.*

A Formalization of Objective and Subjective Time Ontologies

Philip H.P. Nguyen¹

Dan R. Corbett²

¹Justice Technology Services, Department of Justice, Government of South Australia
30, Wakefield Street, Adelaide, SA 5000, Australia

Email: nguyen.philip@saugov.sa.gov.au

²Schafer Corporation

3811, N. Fairfax Drive, Arlington, Va., USA

Email: daniel.corbett.ctr@darpa.mil

Abstract

This paper presents a novel formalization of temporal notions and their classification. First, objective and subjective time perceptions are discussed from a philosophical and logical viewpoint. Then, all objective and subjective temporal concept types are identified, based on McTaggart's A- and B-series and Priorean tense logic, to which temporal events (or propositions) could be mapped. Time ontology is then defined according to a formalism previously introduced by the authors, together with a graphical representation of the proposed temporal concept type hierarchy. Temporal axioms and properties are finally identified, linking our logic with propositional logic.

Keywords: Knowledge Representation, Ontology, Propositional Calculus, Temporal Logic.

1 Introduction

A time ontology is an ontology based on temporal notions. According to current sciences and philosophies, especially of Eastern origin, all objects and phenomena in the universe, whether they are humans, animals, plants, rocks, or a beautiful sunset, are transient, that is, they only exist within a certain timeframe. Since ontology, in its original definition, is a study of reality or existence of "things", it ensues that time is intrinsically part of any ontology. In addition, since temporal notions are sometimes born from subjective perceptions, a time ontology could include elements that are only valid to an individual, a group of individuals, or within a particular context. This paper attempts to formalize time ontology based on objective as well as subjective perceptions, drawing inspirations from logicians such as J.E. McTaggart, A.N. Prior, C. Lejewski, and others. Their theories are still considered valid nowadays, although recent developments have contributed to better formalization of temporal reasoning. The concepts of subjective and objective times have been discussed by philosophers but we believe that this paper presents for

the first time a way to formalize them in an ontology. Our aim is to define an upper time ontology that could be later used in specific applications, such as to describe the temporal content of web pages or to build automated natural language translation engines.

Temporal logic is considered founded by A.N. Prior (1914-1969) (Lejewski 1959). His work and the history of time ontology are detailed by Øhrstrøm and Schärfe (2004). One of the early attempts to formalize time was undertaken by J.F. Allen (1984) with the introduction of a general theory of action and time, in which are categorized time-related actions, such as concurrent actions and their interactions, causation, intention, belief and plan, etc. Causal reasoning is also later expanded in other work (Stein and Morgenstern 1994). More recently, OWL-Time (<http://www.isi.edu/~pan/OWL-Time.html>), formerly DAML-Time, is a project aiming to develop a representative ontology of time that expresses temporal concepts and properties common to any formalization of time, and specifically, the temporal content of web pages and the temporal properties of web services (Hobbs et al. 2004). In OWL-Time, instant and interval are the only two main temporal entities, all other temporal notions being relations over these entities. Time ontologies formalizing instant and interval are also proposed by other authors (Zhou and Fikes 2000). It is interesting to note that Allen (1984) only accepts the concept of time interval but not that of instant or time-point, being considered instead as a "small interval" instead. Our formalization presupposes the concept of instant but does not explicitly elaborate that of interval, which we consider subsumed in the concepts of instant, time direction (i.e., the order between instants), and time continuity or density (i.e., the existence of other instants between any two instants). Other authors, such as Bittner (2002), embody those notions in the definition of a "time-line", which is isomorphic to the set of real numbers, of which a subset is a time interval. OWL-Time permits measurement (or quantification) of time, in terms of temporal unit, calendar and clock, although the concepts of present, past and future are only briefly discussed. OWL-Time could however be considered as an upper ontology on which other more specific or more detailed temporal ontologies (including ours) could be built.

This paper is organized as follows: Section 2 recalls the temporal notions introduced by McTaggart. Section 3 re-formalizes Priorean tense logic with linkages to McTaggart's notions and our proposed temporal concepts of objective and subjective times. Section 4 details our

Copyright © 2007, Australian Computer Society, Inc. This paper appeared at *3rd Australasian Ontology Workshop (AOW-07)*, Gold Coast, Queensland, Australia. Conferences in Research and Practice in Information Technology, Vol. 85. Editors, Thomas Meyer and Abhaya C. Nayak. Reproduction for academic, not-for profit purposes permitted provided this text is included.

time ontology formalization. Section 5 identifies all temporal concept types and represents them in a tree-like hierarchy to assist with understanding. Section 6 details some temporal axioms that are fundamental to our theory, and derives temporal properties that relate our logic to propositional logic. And finally, Section 7 concludes our paper.

2 McTaggart Temporal Concepts

An important aspect of time is the question of its reality, first raised by J.E. McTaggart (1866-1925) in a 1908 article, in which the logician defines three categories of temporal notions: A-series, consisting of notions of past, present and future, B-series, with notions of “earlier than” and “later than” (in fact one notion can be deduced from the other), and C-series, which is B-series without embedded A-series (as we shall see, a B-series always implicitly embeds an A-series). McTaggart maintains that changes are only possible with an A-series, since, in an A-series, any present event was future (at a time in the past) and will be past (at a time in the future), while, in a B-series, if an event M is earlier than another event N, then M will forever be earlier than N, and thus there cannot be any changes in a B-series. On the other hand, from a psychological perspective, without changes, a human mind cannot form any notion of time. Therefore, a B-series cannot be used to define time, only an A-series can. However, a B-series cannot exist without an A-series because one cannot deny that changes must have occurred in order to affirm that an event is earlier or later than another in a B-series (e.g., two events may have occurred at the same location, thus implying that changes must have happened there). Suppose that there is a B-series without an embedded A-series (called a C-series in this case), that series can give us an idea about the order between the events in the series but cannot enable us to form a notion about what *time direction* really means. It is like being presented with two statements made at two different instants, say, in 1995 and in 2000. We know from the order between those two numbers that one statement is made *before* the other, but we cannot know which year we are in now and whether time progresses from 1995 to 2000, or the other way around. In fact, direction is the main characteristic of time and consequently C-series cannot be considered as an appropriate representation of time. So, if an A-series is essential to define time, it follows then that, if an A-series cannot be defined, neither can be time. The difficulty with an A-series is that it is impossible to accurately (i.e., logically or mathematically) define the “present” (or the “now”). Mathematically, a point in time could be defined as the convergence of a series of time intervals, one strictly enclosing the next, with the first time interval enclosing the present moment by some significant margin such that all observers can agree to. For example, if the current time is about 8:00 AM, the first time interval could be from 7:00 AM to 9:00 AM, and the next time interval could be from 7:30 AM to 8:30 AM, and so on. Since it would take an infinity of steps to converge, any “point in time”, in particular the present, can only be a fictitious concept in the mathematical realm. Furthermore, since the present is not static (i.e., time is

always “moving”), at some stage, it is impossible to objectively know whether the time interval being considered in the previous series still contains the present. Therefore, according to McTaggart and mathematical reasoning, time, in particular the present, is not *real*. This is also in line with quantum physics, according to which the existence or reality of a matter and the measurement of time could be quite subjective, although the perception of the present could be experienced by all human beings, with everyone generally being able to consciously perceive what he/she thinks of the very *present moment*.

In summary, the notion of time is subjective, or, at best, can only be considered as relatively objective, i.e., it is only objectively agreed to within certain contexts or bounds. Since anything *subjective* cannot be considered as *real* in the traditional science of physics, McTaggart’s A-series is not real. And so is B-series as B-series implicitly includes A-series. While C-series may be real (since it can be objectively agreed to by all, e.g., no-one can deny that World War I happened *before* World War II), but, as discussed, it cannot be considered as an adequate representation of time. Therefore, any true ontology always relies on subjective temporal notions, whether explicitly or implicitly. In the following, whenever we refer to objectivity in temporal notions, we always mean objectivity in a relative sense, as absolute objectivity cannot be logically proven with time.

Furthermore, in modern logic, an event could be defined as an activity that involves an outcome (Allen 1984). It usually (but not necessarily) has two main attributes: a location and a time (Hobbs et al. 2004). However, in its general definition, an event is “something that happens at a given place and time” (as per <http://wordnet.princeton.edu/>). This means that an event is a record of some changes that occur at some place during some time. Stated differently, event is a result of perception of changes, which also gives rise to the notion of time. Event as change perception therefore precedes the formation of the notion of time (of that event). Event is real (as it can be objectively agreed to) while time is abstract. Thus, event defines time, and in turn, time is used to record event. This is why in our ontological formalization presented in this paper, event and time are closely linked, while in other theories (such as OWL-Time), they may be quite separate.

3 Objective and Subjective Time Ontologies

3.1 Objective Temporal Notions

A.N. Prior defines four “first-grade” temporal notions to express the ideas of “earlier” and “later”, and their qualifications of *temporariness* and *permanency* (Øhrstrøm and Schärfe 2004). We propose to formalize those notions as four concept types, each of them is a function between (\mathbf{PxT}) and \mathbf{P} , where \mathbf{T} is the Time Space and \mathbf{P} is the Proposition Space, as used in propositional calculus (Klement 2006). The four functions could be defined as follows (where $T(p,t)$ means “proposition p is true at instant t”):

- (1) Anteriority (A): $A(p,t) \equiv_{\text{def}} \exists t' \leq t T(p,t')$ (paraphrase: p is true at time t or before)

- (2) Posteriority (Po): $Po(p,t) \equiv_{\text{def}} \exists t' \geq t \ T(p,t')$
(paraphrase: p is true at time t or after)
- (3) Permanent Anteriority (PeA): $PeA(p,t) \equiv_{\text{def}} \forall t' \leq t \ T(p,t')$ (paraphrase: p is always true at time t or before)
- (4) Permanent Posteriority (PePo): $PePo(p,t) \equiv_{\text{def}} \forall t' \geq t \ T(p,t')$ (paraphrase: p is always true at time t or after)

Our definitions above rely on the notions of instant (t), time order (or time direction, which is the order relation " \leq " between two instants), truth of a proposition at an instant ($T(p,t)$), and first-order logic (i.e., the universal and existential quantifiers " \forall " and " \exists "). These four definitions also formalize McTaggart's B-series notion. In addition, to complete A.N. Prior's first-grade notions, three further temporal notions could be derived from the above to express the ideas of temporariness and permanency. These three additional notions are independent of specific instants and are functions between **P** and **P**:

- (5) Temporariness (T) = Anteriority or Posteriority, i.e., $T(p) \equiv_{\text{def}} \exists t \ A(p,t) \cup Po(p,t) = \exists t \ T(p,t)$ (paraphrase: p is temporarily (or sometime) true)
- (6) Permanency (Pe) = Permanent Anteriority and Permanent Posteriority, i.e., $Pe(p) \equiv_{\text{def}} \forall t \ PeA(p,t) \cap PePo(p,t) = \forall t \ T(p,t)$ (paraphrase: p is permanently (or always) true)
- (7) Discrete Permanency = Anteriority and Posteriority, i.e., $DPe(p) \equiv_{\text{def}} \forall t_0 \ A(t_0,p) \cap Po(t_0,p) = \forall t_0 \exists t \exists t' : t \leq t_0 \leq t', T(p,t) \cap T(p,t')$ (paraphrase: p is permanently and discretely true, i.e., at any moment, p is true before and after that moment. This is a new notion first introduced in this paper.)

A.N. Prior's second-grade temporal notions are first-grade notions, plus the notion of the *present* (or the *now*). In fact, Prior's second-grade notions are simply a more explicit expression of first-grade notions, if we accept McTaggart's argument that a B-series always implicitly embeds an A-series. We can now derive from the notion of the present four additional temporal notions. These are independent of specific instants and are functions between **P** and **P**:

- (8) Future (F): $F(p) \equiv_{\text{def}} \exists t \geq \text{Now} \ T(p,t)$ (paraphrase: p will sometime be true)
- (9) Past (Pa): $Pa(p) \equiv_{\text{def}} \exists t \leq \text{Now} \ T(p,t)$ (paraphrase: p was sometime true)
- (10) Permanent Future (PeF): $PeF(p) \equiv_{\text{def}} \forall t \geq \text{Now} \ T(p,t)$ (paraphrase: p will always be true)
- (11) Permanent Past (PePa): $PePa(p) \equiv_{\text{def}} \forall t \leq \text{Now} \ T(p,t)$ (paraphrase: p was always true)

The above 11 notions cover all objective temporal notions in our theory, which also encompass McTaggart's A- and B-series notions and A.N. Prior's first- and second-grade temporal notions.

Based on the above formal definitions, we can easily prove the following properties:

- Temporariness = Future or Past, i.e., $T(p) = F(p) \cup Pa(p)$ (paraphrase: p is temporarily true = p was or will be true)
- Permanency = Permanent Future and Permanent Past, i.e., $Pe(p) = PeF(p) \cap PePa(p)$ (paraphrase: p is

permanently true = p was always and will always be true)

- Discrete Permanency subsumes Future and Past (see formal definition of the subsumption relation in Sect. 4), i.e., $DPe(p) > F(p) \cap Pa(p) = \exists t \exists t' : t \leq \text{Now} \leq t', T(p,t) \cap T(p,t')$ (paraphrase: if p is permanently and discretely true, then in particular, p was true sometime in the past and will be true again sometime in the future).

3.2 Subjective Temporal Notions

The above temporal notions are *objective* as they imply that the time direction between two events ordered by the relation " \leq " could be objectively perceived by all observers. However, as discussed earlier, time is subjective and therefore subjective temporal notions could be formally introduced as follows.

Subjective first-grade temporal notions are defined as functions between the domain set of **P**, **P** \times **T**, or **P** \times **T** \times **O** (where **O** is the observer space), and the value set of **P**: (In the following, $T(p,t,O)$ means "proposition p is true at time t according to observer O".)

- (1) Subjective Anteriority (SA): $SA(p,t,O) \equiv_{\text{def}} \exists t' \leq t \ T(p,t',O)$ (paraphrase: p is true at time t or sometime before, according to observer O)
- (2) Indeterminate Subjective Anteriority (ISA): $ISA(p,t) \equiv_{\text{def}} \exists t' < t \exists O \ T(p,t',O) = \exists O \ SA(p,t,O)$ (paraphrase: p is true at time t or sometime before, according to some observer)
- (3) Subjective Permanent Anteriority (SPeA): $SPeA(p,t,O) \equiv_{\text{def}} \forall t' \leq t \ T(p,t',O)$ (paraphrase: p is always true at time t and before, according to observer O)
- (4) Indeterminate Subjective Permanent Anteriority (ISPeA): $ISPeA(p,t) \equiv_{\text{def}} \forall t' \leq t \exists O \ T(p,t',O)$ (paraphrase: p is always true at time t and before, according to some observers – Note: there may be different observers at different times, i.e., $ISPeA(p,t) \neq \exists O \ SPeA(p,t,O)$)
- (5) Subjective Posteriority (SPo): $SPo(p,t,O) \equiv_{\text{def}} \exists t' \geq t \ T(p,t',O)$ (paraphrase: p is true at time t or sometime after, according to observer O)
- (6) Indeterminate Subjective Posteriority (ISPo): $ISPo(p,t) \equiv_{\text{def}} \exists t' \geq t \exists O \ T(p,t',O) = \exists O \ SPo(p,t,O)$ (paraphrase: p is true at time t or sometime after, according to some observer)
- (7) Subjective Permanent Posteriority (SPePo): $SPePo(p,t,O) \equiv_{\text{def}} \forall t' \geq t \ T(p,t',O)$ (paraphrase: p is always true at time t and after, according to observer O)
- (8) Indeterminate Subjective Permanent Posteriority (ISPePo): $ISPePo(p,t) \equiv_{\text{def}} \forall t' \geq t \exists O \ T(p,t',O)$ (paraphrase: p is always true at time t and after, according to some observers - Note: there may be different observers at different time, i.e., $ISPePo(p,t) \neq \exists O \ SPePo(p,t,O)$)
- (9) Subjective Permanency (SPe): $SPe(p,O) \equiv_{\text{def}} \forall t \ T(p,t,O)$ (paraphrase: p is always true according to observer O)

- (10) Indeterminate Subjective Permanency (ISPe):
 $ISPe(p) \equiv_{\text{def}} \forall t \exists O T(p,t,O)$ (paraphrase: p is always true according to some observers – Note: there may be different observers at different times, i.e., $ISPe(p) \neq \exists O SPe(p,O)$)
- (11) Subjective Temporariness (ST): $ST(p,O) \equiv_{\text{def}} \exists t T(p,t,O)$ (paraphrase: p is sometime true according to observer O)
- (12) Indeterminate Subjective Temporariness (ST):
 $IST(p) \equiv_{\text{def}} \exists t \exists O T(p,t,O) = \exists O ST(p,O)$
 (paraphrase: p is sometime true according to some observer)

In the above, the notion of *subjectivity* expresses the idea that something is true according to one known observer while the notion of *indeterminate subjectivity* conveys that something is true according to some observer or observers, who are only known in particular contexts or particular instants of that observation.

When the notion of the present is added, we can define additional *subjective second-grade temporal notions* similarly:

- (13) Subjective Future (SF): $SF(p,O) \equiv_{\text{def}} \exists t \geq \text{Now} T(p,t,O)$ (paraphrase: p is or will sometime be true according to observer O)
- (14) Indeterminate Subjective Future (ISF): $ISF(p) \equiv_{\text{def}} \exists t \geq \text{Now} \exists O T(p,t,O) = \exists O SF(p,O)$ (paraphrase: p is or will sometime be true according to some observer)
- (15) Subjective Permanent Future (SPeF): $SPeF(p,O) \equiv_{\text{def}} \forall t \geq \text{Now} T(p,t,O)$ (paraphrase: p is and will always be true according to observer O)
- (16) Indeterminate Subjective Permanent Future (ISPeF):
 $ISPeF(p) \equiv_{\text{def}} \forall t \geq \text{Now} \exists O T(p,t,O)$ (paraphrase: p is and will always be true according to some observers - Note: there may be different observers at different times, i.e., $ISPeF(p) \neq \exists O SPeF(p,O)$)
- (17) Subjective Past (SPa): $SPa(p,O) \equiv_{\text{def}} \exists t \leq \text{Now} T(p,t,O)$ (paraphrase: p is or was sometime true according to observer O)
- (18) Indeterminate Subjective Past (ISPa): $ISPa(p) \equiv_{\text{def}} \exists t \leq \text{Now} \exists O T(p,t,O) = \exists O SPa(p,O)$ (paraphrase: p is or was sometime true according to some observer)
- (19) Subjective Permanent Past (SPePa): $SPePa(p,O) \equiv_{\text{def}} \forall t \leq \text{Now} T(p,t,O)$ (paraphrase: p is and was always true according to observer O)
- (20) Indeterminate Subjective Permanent Past (ISPePa):
 $ISPePa(p) \equiv_{\text{def}} \forall t \leq \text{Now} \exists O T(p,t,O)$ (paraphrase: p is and was always true according to some observers - Note: there may be different observers at different times, i.e., $ISPePa(p) \neq \exists O SPePa(p,O)$)
- (21) Subjective Discrete Permanency = Subjective Anteriority and Subjective Posteriority: $SDPe(p,O) \equiv_{\text{def}} (\forall t_0 \exists O SA(p,t_0,O) \cap SPO(p,t_0,O)) = (\forall t_0 \exists t \exists t': t \leq t_0 \leq t', T(p,t,O) \cap T(p,t',O))$ (paraphrase: At any instant t, p is true before and after t according to observer O, in particular, p was true and will be true again according to observer O). Note that Subjective Discrete Permanency subsumes Subjective Past and Subjective Future, i.e., $SDPe(p,O) > (SPa(p,O) \cap SF(p,O))$ since the right part of the equation is equal to $(\exists t \exists t': t \leq \text{Now} \leq t', T(p,t,O) \cap T(p,t',O))$

- (22) Indeterminate Subjective Discrete Permanency = Indeterminate Subjective Anteriority and Indeterminate Subjective Posteriority, i.e., $ISDPe(p) \equiv_{\text{def}} (\forall t_0 \exists O \exists O' SA(p,t_0,O) \cap SPO(p,t_0,O')) = (\forall t_0 \exists O \exists O' \exists t \exists t': t \leq t_0 \leq t', T(p,t,O) \cap T(p,t',O'))$ (paraphrase: if p is discretely and permanently true according some observers, then in particular, p was true and will be true again according to some observers – Note: there may be different observers at different times, i.e., $ISDPe(p) \neq \exists O SDPe(p,O)$). Also note that Indeterminate Subjective Discrete Permanency subsumes Indeterminate Subjective Past and Indeterminate Subjective Future, i.e., $ISDPe(p) > (\exists O \exists O' SPa(p,O) \cap SF(p,O'))$ since the right part of the equation is equal to $(\exists t \exists t' \exists O \exists O': t \leq \text{Now} \leq t', T(p,t,O) \cap T(p,t',O'))$

The above 22 definitions cover all subjective temporal notions in our formalism, which also extend McTaggart's A- and B-series notions and A.N. Prior's first- and second-grade temporal notions, into subjectivity.

4 Proposed Ontology Formalization

Nguyen and Corbett (2003, 2006) define an *ontology* as a semantically consistent subset of a *canon*, which is in essence a mapping of a real world onto an abstract world. In this paper, to simplify and without loss of generality, we consider these two notions identical.

In our formalism, a *time ontology* (or *time canon*) could be formally defined as a 5-tuple $K = (T, I, <, conf, B)$ in which:

- (1) T is the set of temporal concept and relation types, i.e., $T = T_C \cup T_R$ where:
 - (a) T_C is the set of temporal concept types, consisting of 11 objective and 22 subjective temporal notions as listed above.
 - (b) T_R is the set of temporal relation types, consisting of 3 elements similar to the three main logical connectives of propositional calculus, i.e., negation (\neg), conjunction (\cap), and disjunction (\cup) (Smith 2003), defined as follows:
 - \neg is a unary relation over T_C , i.e. $\neg: T_C \rightarrow T_C$ with $\forall c \in T_C$ the value $\neg(c)$ (simply written as $\neg c$) is a temporal concept type defined over the same domain set as c , i.e., $\forall p \in \mathbf{P} \forall t \in \mathbf{T} \forall O \in \mathbf{O}$
 - if c is defined over \mathbf{P} only, then $(\neg c)(p) = \neg(c(p))$
 - if c is defined over $\mathbf{P} \times \mathbf{T}$, then $(\neg c)(p,t) = \neg(c(p,t))$
 - if c is defined over $\mathbf{P} \times \mathbf{T} \times \mathbf{O}$, then $(\neg c)(p,t,O) = \neg(c(p,t,O))$
 - \cap is a binary relation over $T_C \times T_C$, i.e., $\cap: T_C \times T_C \rightarrow T_C$ with $\forall c, c' \in T_C$ the value $\cap(c,c')$ (simply written as $c \cap c'$) is a temporal concept type defined over the largest of the 2 domain sets used by c and c' , i.e.,

- if c and c' are both defined over \mathbf{P} only, or over \mathbf{PxT} , or over \mathbf{PxTxO} , then so is $c \cap c'$ with: $\forall p \in \mathbf{P} \forall t \in \mathbf{T} \forall O \in \mathbf{O}$
 - $(c \cap c')(p) = c(p) \cap c'(p)$
 - or $(c \cap c')(p, t) = c(p, t) \cap c'(p, t)$
 - or $(c \cap c')(p, t, O) = c(p, t, O) \cap c'(p, t, O)$
- if there is a difference in the domain sets of c and c' , then $c \cap c'$ is defined over the largest domain set of the two, e.g., if c is defined over \mathbf{P} only and c' is defined over \mathbf{PxTxO} , then $c \cap c'$ is defined over \mathbf{PxTxO} with: $\forall p \in \mathbf{P} \forall t \in \mathbf{T} \forall O \in \mathbf{O}$
 - $(c \cap c')(p, t, O) = c(p) \cap c'(p, t, O)$

- \cup is defined similarly to \cap .

- (2) I is the set of *instances* of temporal concept types in T_C . I consists of all *atomic* propositions that contain temporal notions, i.e., temporal propositions that cannot be further divided into sub-propositions connected by any of the four logical connectives of propositional calculus: “and”, “or”, “not”, and “implication”. For example, the proposition “it was hot yesterday but it will be cooler tomorrow” could be considered as two atomic temporal propositions: “it was hot yesterday” and “it will be cooler tomorrow” connected by the logical connective “and” (i.e., “ \cap ”). Note that our definition of *temporal proposition* is what OWL-Time calls *eventuality* or *event*.
- (3) “ $<$ ” is the subsumption relation in T , defined as a binary relation between temporal concept types or between temporal relation types, such that the first type is *semantically entailed* by the second, e.g., the relation “ $b < a$ ” or “ $a > b$ ” between two temporal concept types a and b means: “ a semantically entails b ”. This subsumption relation is based on the *semantic entailment* relation of propositional calculus (normally represented by the symbol “ \models ”) (Smith 2003). As we shall see, in some cases, *semantic entailment* in our subsumption relation also means *syntactic proof* (normally represented by the symbol “ \vdash ”) (Smith 2003). Formally, “ $<$ ” can be defined as follows:

- Subsumption relation in T_C :

$\forall c, c' \in T_C$ we have: $c > c'$ if and only if:

- a) If the domain set of c' is larger than, or equal to, that of c , then the semantic entailment relation between the propositions transformed by c and c' (i.e., the values of the functions c and c') must be true for all instances of the common domain set (between c and c'), and for all instances of each extra dimension of the domain set of c' , i.e.,

$$\begin{aligned} c > c' &\equiv_{\text{def}} \forall p \in \mathbf{P} \forall t \in \mathbf{T} \forall O \in \mathbf{O} \quad c(p) \models c'(p) \\ &\quad c'(p) \models c'(p, t) \models c'(p, O) \models c'(p, t, O) \\ \text{or } c > c' &\equiv_{\text{def}} \forall p \in \mathbf{P} \forall t \in \mathbf{T} \forall O \in \mathbf{O} \quad c(p, t) \models c'(p, t) \models c'(p, t, O) \\ \text{or } c > c' &\equiv_{\text{def}} \forall p \in \mathbf{P} \forall t \in \mathbf{T} \forall O \in \mathbf{O} \quad c(p, t, O) \models c'(p, t, O) \end{aligned}$$

- b) If the domain set of c is larger than that of c' , then the semantic entailment relation between the propositions transformed by c and c' (i.e., the values of the functions c and c') must be true for all instances of the common domain set (between c and

c'), and for at least one instance of each extra dimension of the domain set of c , i.e.,

$$\begin{aligned} c > c' &\equiv_{\text{def}} \forall p \in \mathbf{P} \exists t \in \mathbf{T} \quad c(p, t) \models c'(p) \\ \text{or } c > c' &\equiv_{\text{def}} \forall p \in \mathbf{P} \exists O \in \mathbf{O} \quad c(p, O) \models c'(p) \\ \text{or } c > c' &\equiv_{\text{def}} \forall p \in \mathbf{P} \exists t \in \mathbf{T} \exists O \in \mathbf{O} \quad c(p, t, O) \models c'(p) \\ \text{or } c > c' &\equiv_{\text{def}} \forall p \in \mathbf{P} \forall t \in \mathbf{T} \exists O \in \mathbf{O} \quad c(p, t, O) \models c'(p, t) \\ \text{or } c > c' &\equiv_{\text{def}} \forall p \in \mathbf{P} \forall O \in \mathbf{O} \exists t \in \mathbf{T} \quad c(p, t, O) \models c'(p, O) \end{aligned}$$

In the above, the symbol “ \models ” means “logical or”, e.g., “ $c(p) \models c(p, t) \models c(p, t, O)$ ” means “ $c(p)$, $c(p, t)$, or $c(p, t, O)$ ”, depending on the domain set of c' . Note that in the above, condition a) is generally used to determine that an objective concept subsumes a subjective concept of the same nature (such as $\text{PeA} > \text{SPeA}$), while condition b) is generally used to determine the subsumption relation between two concepts of the same category (i.e., both objective or both subjective, such as $\text{SA} > \text{ISA}$).

- Subsumption relation in T_R :

The subsumption relation “ $<$ ” among the temporal relation types in T_R could be formally defined as: $\forall r, r' \in T_R \quad r > r' \equiv_{\text{def}} \forall c, c' \in T_C \quad r(c, c') > r'(c, c')$ with the relation “ $r(c, c') > r'(c, c')$ ” defined similarly to the relation “ $<$ ” between two elements of T_C as above. In fact, since there are only 3 temporal relation types: \neg , \cap , and \cup , it can be proven that the only subsumption relation in T_R is: “ $\cap > \cup$ ”. Indeed, $\forall c, c' \in T_C$, we have (assuming that c and c' are defined over \mathbf{P} only, to simplify):

$$\begin{aligned} (c \cap c') &> (c \cup c') \\ \text{or } \forall p \in \mathbf{P} \quad (c \cap c')(p) &\models (c \cup c')(p) \\ \text{or } \forall p \in \mathbf{P} \quad (c(p) \cap c'(p)) &\models (c(p) \cup c'(p)) \end{aligned}$$

The last statement is true because in propositional calculus, “any two propositions that are jointly true always imply that either proposition is true”.

- (4) *conf* is the “conformity” relation, defined between the set of all non-tautological temporal concept type instances (denoted as $\Lambda\{*\}$) and the set of all temporal concept types T_C , i.e., *conf*: $\Lambda\{*\} \rightarrow T_C$ where $\{*\}$ represents the set of all *tautologies* in propositional calculus. The *conf* function expresses the idea that any atomic temporal proposition, except a tautology, can be associated with a temporal concept type. For example, the temporal proposition: “The phenomenon p has been observed throughout the ages” can be translated as “ $\forall t \leq \text{Now} \exists O \quad T(p, t, O)$ ”, or p can be associated with the “Indeterminate Subjective Permanent Past” concept type of T_C (i.e., if we call that statement q , then $q \in \Lambda\{*\}$ and *conf*(q) = ISPePa). We should distinguish that statement with: “Someone has always observed the phenomenon p ”, translated as “ $\exists O \forall t \leq \text{Now} \quad T(p, t, O)$ ”, or p is an instance of the “Subjective Permanent Past” concept type. Similarly, the statement: “The truth p will be revealed to all in the future” could be translated as “ $\exists t \geq \text{Now} \forall O \quad T(p, t, O)$ ”, or “ $\exists t \geq \text{Now} \quad T(p, t)$ ”, or “ p is a Future truth” (i.e., p is an instance of the “Future” temporal concept type), while the statement: “Someone will

know the truth p” could be translated as “ $\exists O \exists t \geq \text{Now } T(p,t,O)$ ”, or “p is a Subjective Future” truth (i.e., p is an instance of the “Subjective Future” temporal concept type). Note that in OWL-Time, the relations between propositions and times are *atTime*(e,t|T) and *holds*(e,t|T) (meaning “the proposition or event e holds at instant t or during interval T”). These relations are similar to our *conf* function. OWL-Time separates the event (or proposition) ontology from the time ontology. (In fact, *atTime* is a relation in the time ontology while *holds* is a relation in the event ontology, although both have the same semantics in OWL-Time.) In our formalism, we link them together because as discussed earlier we consider that propositions and events are part of the real world while time is part of an abstract world, and an ontology is a formal attempt to link those two worlds (Nguyen et al. 2006).

- (5) *B* is the *Canonical Basis* function, defined between T_R and the set of all subsets of T_C (denoted as $\varphi(T_C)$), i.e., $B: T_R \rightarrow \varphi(T_C)$. *B* expresses the “usage pattern” (or “canonical basis”) of each temporal relation type, that is, it defines which temporal concept types can be used in each temporal relation type. In our time ontology, based on the above definitions of T_C and T_R there is no restriction and any temporal concept type can be used with any temporal relation type. This is similar to propositional calculus, in which the relations \neg , \cap , and \cup can be used with any propositions.

Finally, note that our formalism could be considered as a *meta-logic* since it is defined on top of propositional logic.

5 Representation of Time Ontologies

In the objective time ontology, the previously identified 11 objective temporal concepts could be *syntactically proven* to be linked by 10 subsumption relations, based on their predicate formulae specified in Section 3. (More correctly, those 10 relations are 10 *supertypes* (Sowa 1984), as some relations are between more than two concepts.) This means that our temporal subsumption relation (“<”) that is based on *semantic entailment* can also be said to be based on *syntactic proof* (Smith 2003):

1. Anteriority > Temporariness
2. Discrete Permanency > Anteriority, Future, Past, Posteriority
3. Future > Temporariness
4. Past > Temporariness
5. Permanency > Discrete Permanency, Permanent Anteriority, Permanent Future, Permanent Past, Permanent Posteriority
6. Permanent Anteriority > Anteriority
7. Permanent Future > Future
8. Permanent Past > Past
9. Permanent Posteriority > Posteriority
10. Posteriority > Temporariness

Figure 1 (drawn with a tool built by the authors (Nguyen et al. 2006)) shows the objective temporal concept type hierarchy. Note that ‘permanency’ is at the top of the hierarchy while ‘temporariness’ is at its bottom. (Also

note that in all figures, concept names between parentheses are *co-references* (Sowa 1984).)

Similarly, it can be syntactically proven that there are 21 (n-ary) subsumption relations among the 22 subjective temporal concept types, forming a hierarchy represented in Figure 2 (with acronyms used in order to reduce the figure size). Note that ‘subjective permanency’ is at the top of the hierarchy while ‘indeterminate subjective temporariness’ is at its bottom.

In the combined objective-subjective ontology, we can identify additional subsumption relations linking objective with subjective concepts, based on the formal definition of the subsumption relation in Sect. 4. In general, an objective concept semantically entails (or subsumes) the subjective concept of the same nature, since “an objectively true proposition” means “a proposition true to all observers”. Also, no subjective concept type can subsume an objective concept type due to the extra observer dimension needed in the former. Therefore, the following 11 additional subsumption relations forms the complete list of objective-subjective relationships (with acronyms used for legibility):

1. A > SA
2. Po > SPo
3. F > SF
4. Pa > SPa
5. T > ST
6. DPe > SDPe
7. Pe > SPE
8. PeA > SPeA
9. PePo > SPePo
10. PeF > SPeF
11. PePa > SPePa

Finally, if we add the above 11 objective-subjective relations to the previous 10 objective and 21 subjective relations, we obtain a total of 42 relations, which can be consolidated into 32 (n-ary) subsumption relations (after relation consolidation (Nguyen et al. 2006)) between the 33 objective and subjective temporal concept types. They can be fully listed as follows:

1. A > SA, T
2. DPe > A, F, Pa, Po, SDPe
3. F > SF, T
4. ISA > IST
5. ISDPe > ISA, ISF, ISPa, ISPo
6. ISF > IST
7. ISPa > IST
8. ISPe > ISPeA, ISPeF, ISPePa, ISPePo
9. ISPeA > ISA
10. ISPeF > ISF
11. ISPePa > ISPa
12. ISPePo > ISPo
13. ISPo > IST
14. Pa > SPa, T
15. Pe > DPe, PeA, PeF, PePa, PePo, SPE
16. PeA > A, SPeA
17. PeF > F, SPeF
18. PePa > Pa, SPePa
19. PePo > Po, SPePo
20. Po > SPo, T

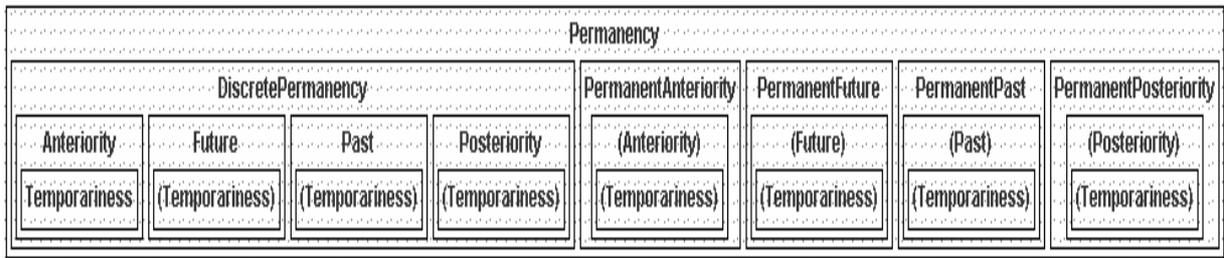


Fig. 1. Objective Temporal Concept Type Hierarchy

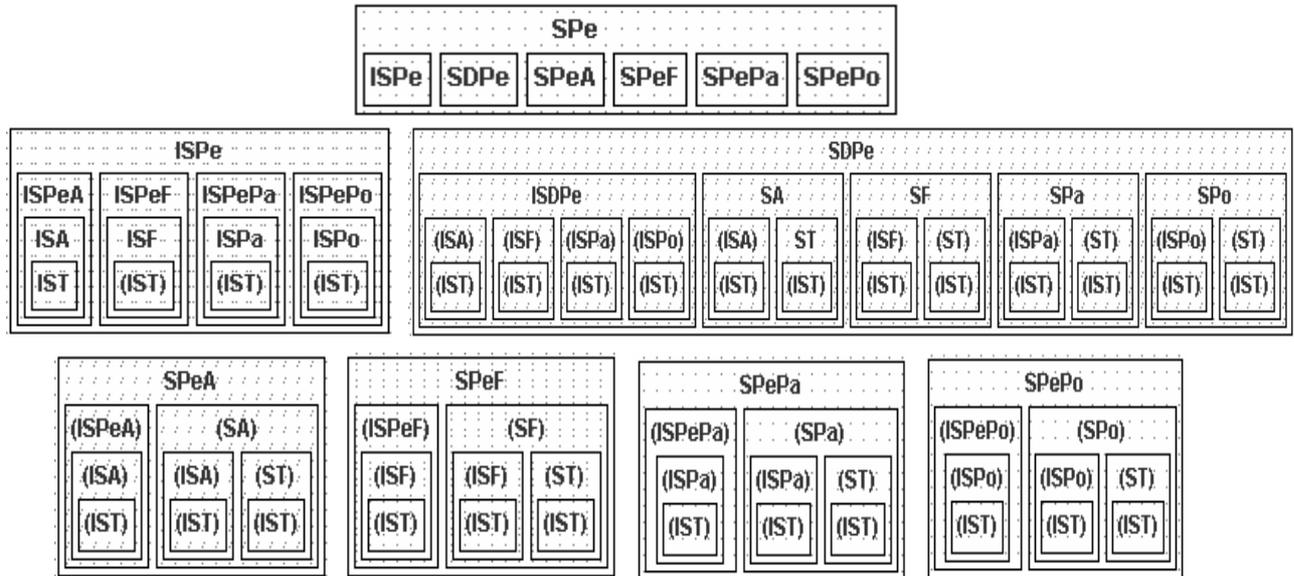


Fig. 2. Subjective Temporal Concept Type Hierarchy

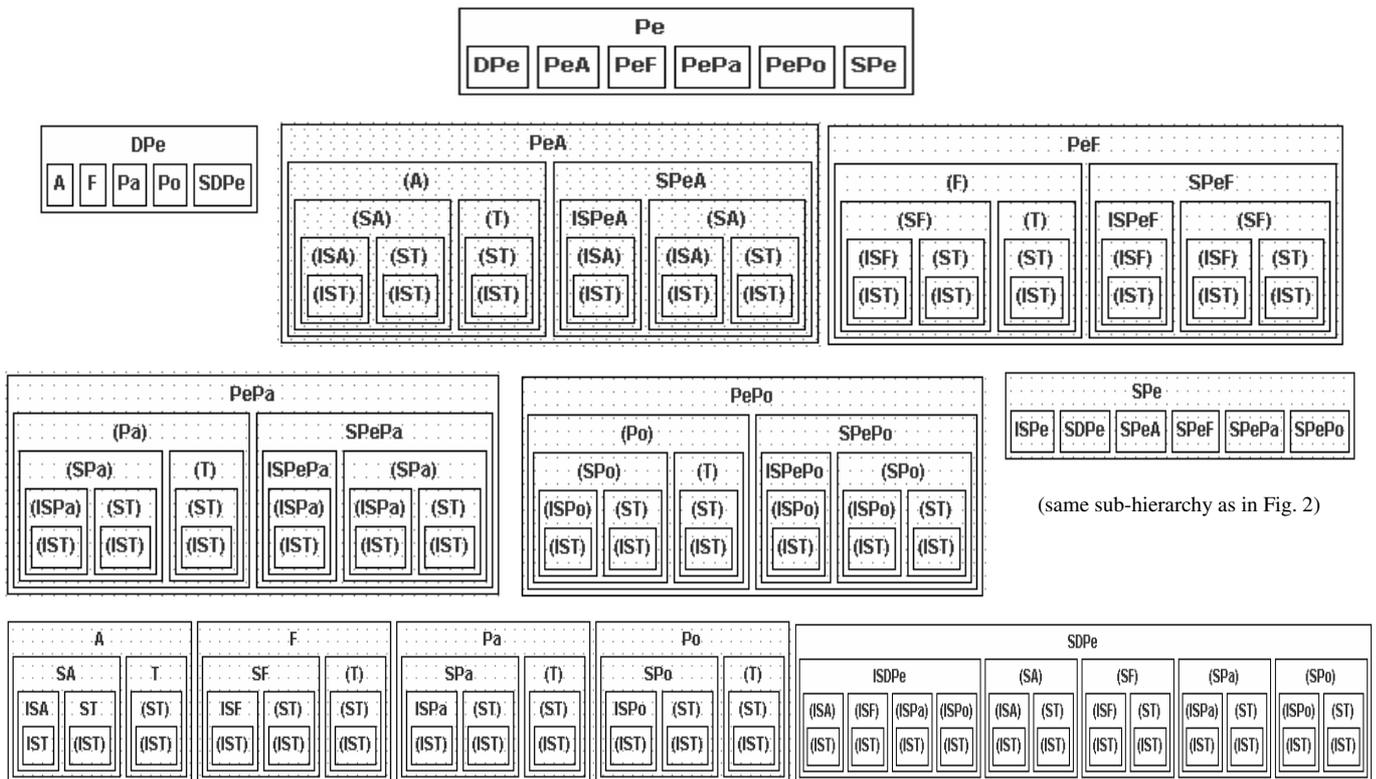


Fig. 3. Combined Temporal Concept Type Hierarchy

21. SA > ISA, ST
22. SDPe > ISDPe, SA, SF, SPa, SPo
23. SF > ISF, ST
24. SPa > ISPa, ST
25. SPe > ISPe, SDPe, SPeA, SPeF, SPePa, SPePo
26. SPeA > ISPeA, SA
27. SPeF > ISPeF, SF
28. SPePa > ISPePa, SPa
29. SPePo > ISPePo, SPo
30. SPo > ISPo, ST
31. ST > IST
32. T > ST

Based on these subsumption relations, the combined temporal concept type hierarchy could be represented in Figure 3. Note that ‘permanency’ (coming from the objective ontology) is at the top of the hierarchy while ‘indeterminate subjective temporariness’ (coming from the subjective ontology) is at its bottom, as one may intuitively expect in light of the earlier remarks on objective-subjective subsumption relations.

6 Temporal Axioms and Properties

In this section, we will attempt to identify key axioms and properties in our temporal logic. We call our axioms *Truth Axioms* because they express the semantics of the truth functions $T(p,t)$ and $T(p,t,O)$.

6.1 Temporal Axioms

- Truth Axiom 1: $\forall p \ p \Rightarrow (\forall c \ c(p))$

(paraphrase: If a proposition is true, then it is true under any temporal concept type.)

This axiom is the most basic and fundamental in our theory. It simply states that if a proposition is true without any temporal qualification, then it is supposed to be *permanently* true. And since it is permanently true and permanency is at the top of our temporal concept type hierarchy, it is true with any subtype of permanency, i.e., true with any other temporal concept type.

- Truth Axiom 2:

$$(2a) \quad T(p \Rightarrow q, t) = (T(p, t) \Rightarrow T(q, t)) \\ \text{and } T(p \Rightarrow q, t, O) = (T(p, t, O) \Rightarrow T(q, t, O))$$

(paraphrase: If at time t (and according to observer O), “ p implies q ” is true, then “ p is true at t (and according to observer O)” implies “ q is true at time t (and according to observer O)”, and vice-versa.)

$$(2b) \quad T(p \wedge q, t) = (T(p, t) \wedge T(q, t)) \\ \text{and } T(p \wedge q, t, O) = (T(p, t, O) \wedge T(q, t, O))$$

(paraphrase: If at time t (and according to observer O), both p and q are true, then both “ p is true at time t (and according to observer O)” and “ q is true at time t (and according to observer O)” are true, and vice-versa.)

$$(2c) \quad T(p \cup q, t) = (T(p, t) \cup T(q, t)) \\ \text{and } T(p \cup q, t, O) = (T(p, t, O) \cup T(q, t, O))$$

(paraphrase: If at time t (and according to observer O), either p or q is true, then either “ p is true at time t (and

according to observer O)” or “ q is true at time t (and according to observer O)” is true, and vice-versa.)

$$(2d) \quad \neg T(p, t) = T(\neg p, t) \\ \text{and } \neg T(p, t, O) = T(\neg p, t, O)$$

(paraphrase: If at time t (and according to observer O), p is not true, then it is true that “ p is not true at time t (and according to observer O)”, and vice-versa.)

- Truth Axiom 3:

$$(3a) \quad (\forall t \ T(p \Rightarrow q, t)) = ((\forall t \ T(p, t) \Rightarrow (\forall t' \ T(q, t')))) \\ \text{and } (\forall t \ \forall O \ T(p \Rightarrow q, t, O)) = ((\forall t \ \forall O \ T(p, t, O) \Rightarrow (\forall t' \ \forall O' \ T(q, t', O'))))$$

(paraphrase: If at any time (and according to any observer), “ p implies q ” is true, then “ p is true at all times (and according to all observers)” implies “ q is true at all times (and according to all observers)”, and vice-versa.)

$$(3b) \quad (\forall t \ T(p \wedge q, t)) = ((\forall t \ T(p, t) \wedge (\forall t' \ T(q, t')))) \\ \text{and } (\forall t \ \forall O \ T(p \wedge q, t, O)) = ((\forall t \ \forall O \ T(p, t, O) \wedge (\forall t' \ \forall O' \ T(q, t', O'))))$$

(paraphrase: If at any time (and according to any observer), both p and q are true, then both “ p is true at all times (and according to all observers)” and “ q is true at all times (and according to all observers)” are true, and vice-versa.)

$$(3c) \quad ((\forall t \ T(p, t) \cup (\forall t' \ T(q, t'))) \Rightarrow (\forall t \ T(p \cup q, t))) \\ \text{and } ((\forall t \ \forall O \ T(p, t, O) \cup (\forall t' \ \forall O' \ T(q, t', O'))) \Rightarrow (\forall t \ \forall O \ T(p \cup q, t, O)))$$

(paraphrase: If either “ p is true at all times (and according to all observers)” or “ q is true at all times (and according to all observers)” is true, then at any time (and according to any observer), either p or q is true.) Note that the converse of this Truth Axiom does **not** hold.

$$(3d) \quad (\forall t \ \neg T(p, t)) = (\forall t \ T(\neg p, t)) \\ \text{and } (\forall t \ \forall O \ \neg T(p, t, O)) = (\forall t \ \forall O \ T(\neg p, t, O))$$

(paraphrase: If at all times (and according to all observers), p is not true, then it is true at all times (and according to all observers) that “ p is not true (at those times (and according to those observers))”, and vice-versa.)

- Truth Axiom 4:

$$(4a) \quad (\forall t_0 \ \exists t \ \exists t': t \leq t_0 \leq t', T(p, t) \wedge T(p, t') \Rightarrow T(q, t) \wedge T(q, t')) \\ \Rightarrow ((\forall t_0 \ \exists t \ \exists t': t \leq t_0 \leq t', T(p, t) \wedge T(p, t')) \Rightarrow (\forall s_0 \ \exists s \ \exists s': s \leq s_0 \leq s', T(q, s) \wedge T(q, s'))))$$

and

$$(\forall O \ \forall t_0 \ \exists t \ \exists t': t \leq t_0 \leq t', T(p, t, O) \wedge T(p, t', O) \Rightarrow T(q, t, O) \wedge T(q, t', O)) \Rightarrow$$

$$((\forall O \ \forall t_0 \ \exists t \ \exists t': t \leq t_0 \leq t', T(p, t, O) \wedge T(p, t', O)) \Rightarrow (\forall O' \ \forall s_0 \ \exists s \ \exists s': s \leq s_0 \leq s', T(q, s, O') \wedge T(q, s', O')))$$

(paraphrase: If at any time t (and according to any observer), “ p is true before and after t ” implies “ q is true before and after t (but at the same times as p)”, then “ p is true before and after t , at any time t (and according to any observer)” implies “ q is true before and after s , at any time s (and according to any observer) (the times before

and after s could be different from those relating to p”). Note that the converse of this Truth Axiom does **not** hold.

$$(4b) (\forall t_0 \exists t \exists t': t \leq t_0 \leq t', (T(p,t) \wedge T(p,t')) \wedge (T(q,t) \wedge T(q,t'))) \Rightarrow ((\forall t_0 \exists t \exists t': t \leq t_0 \leq t', T(p,t) \wedge T(p,t')) \wedge (\forall s_0 \exists s \exists s': s \leq s_0 \leq s', T(q,s) \wedge T(q,s')))$$

and

$$(\forall O \forall t_0 \exists t \exists t': t \leq t_0 \leq t', (T(p,t,O) \wedge T(p,t',O)) \wedge (T(q,t,O) \wedge T(q,t',O))) \Rightarrow ((\forall O \forall t_0 \exists t \exists t': t \leq t_0 \leq t', T(p,t,O) \wedge T(p,t',O)) \wedge (\forall O' \forall s_0 \exists s \exists s': s \leq s_0 \leq s', T(q,s,O') \wedge T(q,s',O')))$$

(paraphrase: If at any time t (and according to any observer), both “p is true before and after t” and “q is true before and after t (but at the same times as p)”, then both “p is true before and after t, at any time t (and according to any observer)” and “q is true before and after s, at any time s (and according to any observer) (the times before and after s could be different from those relating to p)”). Note that the converse of this Truth Axiom does **not** hold.

$$(4c) ((\forall t_0 \exists t \exists t': t \leq t_0 \leq t', T(p,t) \wedge T(p,t')) \cup (\forall s_0 \exists s \exists s': s \leq s_0 \leq s', T(q,s) \wedge T(q,s'))) \Rightarrow (\forall t_0 \exists t \exists t': t \leq t_0 \leq t', (T(p,t) \wedge T(p,t')) \cup (T(q,t) \wedge T(q,t')))$$

and

$$((\forall O \forall t_0 \exists t \exists t': t \leq t_0 \leq t', T(p,t,O) \wedge T(p,t',O)) \cup (\forall O' \forall s_0 \exists s \exists s': s \leq s_0 \leq s', T(q,s,O') \wedge T(q,s',O'))) \Rightarrow (\forall O \forall t_0 \exists t \exists t': t \leq t_0 \leq t', (T(p,t,O) \wedge T(p,t',O)) \cup (T(q,t,O) \wedge T(q,t',O)))$$

(paraphrase: If either “p is true before and after t, at any time t (and according to any observer)” or “q is true before and after s, at any time s (and according to any observer)” is true, then at any time t (and according to any observer), either “p is true before and after t” or “q is true before and after t” is true.) Note that the converse of this Truth Axiom does **not** hold.

$$(4d) (\forall t_0 \exists t \exists t': t \leq t_0 \leq t', \neg(T(p,t) \wedge T(p,t'))) = (\forall t_0 \exists t \exists t': t \leq t_0 \leq t', T(\neg p,t) \vee T(\neg p,t'))$$

and

$$(\forall O \forall t_0 \exists t \exists t': t \leq t_0 \leq t', \neg(T(p,t,O) \wedge T(p,t',O))) = (\forall O \forall t_0 \exists t \exists t': t \leq t_0 \leq t', T(\neg p,t,O) \vee T(\neg p,t',O))$$

(paraphrase: If at any time t (and according to any observer), it is not true that we have both “p is true before t” and “p is true after t”, then at any time t (and according to any observer), either “p is not true before t” or “p is not true after t” is true. The converse also holds.) Note that this axiom is only added for completeness, as it is simply a deduction of De Morgan’s theorem in propositional calculus and the above Truth Axiom 2d.

6.2 Temporal Properties

Based on the above Truth Axioms, the following properties linking our temporal formalization and propositional logic could be proven syntactically.

In the following, we use the symbol ‘ \Rightarrow ’ to denote the implication relation in propositional calculus and also, to simplify the notations we will suppose that c is defined over **P** only, as similar properties could be written when c is defined over **PxT** or **PxTxO**.

For any temporal concept type c in T_C and for any propositions p, q and r in **P**, we have the following properties:

$$(1) c(p \Rightarrow q) \models (c(p) \Rightarrow c(q))$$

(paraphrase: if “p implies q” is true under a temporal concept type c, then “p is true under c” implies “q is true under c”). For example, if the proposition: “p implies q” has always been true (i.e., the proposition is a Permanent Past truth), then the proposition: “p has always been true” implies the proposition: “q has always been true”.

$$(2) (p \Rightarrow q) \models (c(p) \Rightarrow c(q))$$

(paraphrase: if “p implies q” is true, then for any temporal concept type c, “p is true under c” implies “q is true under c”).

$$(3) c(\neg p) \models \neg c(p)$$

(paraphrase: For a temporal concept type c, if “non-p is true under c”, then it is not true that “p is true under c”). Note that the converse of this property does **not** hold.

$$(4) c(p \wedge q) \models (c(p) \wedge c(q))$$

(paraphrase: If the proposition “p and q are true” is true under c (i.e., p and q are jointly true under c), then both propositions: “p is true under c” and “q is true under c” are true.) For example, if both p and q will always be jointly true (i.e., “p and q” is a Permanent Future truth), then “p will always be true” and “q will always be true” are both true.

$$(5) (c(p) \cup c(q)) \models c(p \cup q)$$

(paraphrase: if either proposition “p is true under c” and “q is true under c” is true, then the proposition “either p or q is true” is true under c.) For example, if either “p will always be true” or “q will always be true” is true, then “p or q is true” will always be true. Note that the converse of this property is not true, e.g., if “p or q is true” is a Discrete Permanent truth, then it is not necessarily true that either “p is a Discrete Permanent truth” or “q is a Discrete Permanent truth” is true, since, at any time t, “p or q” is true before and after t (e.g., p is true before t and q is true after t), but it is not necessarily true that “p is true both before and after t” or “q is true both before and after t”.

$$(6) \neg(c(p) \wedge c(q)) = (\neg c(p) \cup \neg c(q))$$

This is an extension of De Morgan’s Theorem No. 1 in propositional calculus.

(paraphrase: If it is not true that both “p true under c” and “q true under c” can be jointly true, then it must be true that either “p not true under c” or “q not true under c” is true, and vice-versa.) For example, if we cannot have both “p is a Discrete Permanent truth” and “q is a Discrete Permanent truth”, then either “p is not a Discrete Permanent truth” or “q is not a Discrete Permanent truth” is true, and vice-versa.

$$(7) (c(\neg p) \cup c(\neg q)) \models \neg(c(p) \wedge c(q))$$

This is an extension of Temporal Property 6 above.

$$(8) \neg(c(p) \cup c(q)) = (\neg c(p) \wedge \neg c(q))$$

This is an extension of De Morgan’s Theorem No. 2 in propositional calculus.

(paraphrase: If it is not true that either “p true under c” or “q true under c” is true, then it is true that “p not true under c” and “q not true under c” are both true.) The converse also holds. For example, if we cannot have either p or q as a Future truth, then we can have both “p

not a Future truth” and “q not a Future truth” (i.e., both p and q are not Future truths), and vice-versa.

$$(9) (c(\neg p) \wedge c(\neg q)) \vdash \neg(c(p) \vee c(q))$$

This is an extension of Temporal Property 8 above.

(10) Temporal Modus Ponens: $c((p \Rightarrow q) \wedge p) \vdash c(q)$ (paraphrase: If both “p implies q” and p are true under c, then q is true under c.) Note that it could be proven that a similar Temporal Modus Ponens formula does **not** hold:

$$(c(p \Rightarrow q) \wedge c(p)) \not\vdash c(q)$$

(11) Temporal Modus Tollens:

$$c((p \Rightarrow q) \wedge \neg q) \vdash c(\neg p)$$

(paraphrase: If both “p implies q” and “not q” are true under c, then “not p” is true under c.) Note that it could be proven that the similar following Temporal Modus Tollens formulae do **not** hold:

$$- c((p \Rightarrow q) \wedge \neg q) \not\vdash \neg c(p)$$

$$- (c(p \Rightarrow q) \wedge \neg c(q)) \not\vdash c(\neg p)$$

$$- (c(p \Rightarrow q) \wedge \neg c(q)) \not\vdash \neg c(p)$$

(12) Temporal Transposition:

$$c(p \Rightarrow q) \vdash (\neg c(q) \Rightarrow \neg c(p))$$

(paraphrase: If “p implies q” is true under c, then it is true that “q not true under c” implies “p not true under c”.) Note that Transposition is similar to Modus Tollens, but they are not the same as Modus Tollens emphasizes the non-true value of q in the conclusion while Transposition emphasizes the semantic entailment relation in the conclusion proposition.

(13) Temporal Distribution:

$$(c(p) \vee c(q \wedge r)) \vdash (c(p \vee q) \wedge c(p \vee r))$$

(paraphrase: If “p true under c” or “q and r jointly true under c”, then both “p or q true under c” and “p or r true under c” are true.) Note that it could be proven that the following similar formulae do **not** hold:

$$- (c(p) \wedge c(q \vee r)) \not\vdash (c(p \wedge q) \vee c(p \wedge r))$$

$$- (c(p \vee q) \wedge \neg c(p)) \not\vdash c(q)$$

$$- (c(p \Rightarrow q) \wedge c(q \Rightarrow r)) \not\vdash c(p \Rightarrow r)$$

We apologize for not being able to include the proofs for the above properties in this paper due to space restriction.

7 Conclusion

This paper proposes a novel formalization of temporal notions in which all objective and subjective temporal concept types are identified, based on McTaggart’s A- and B-series and Priorean tense logic, with the help of propositional calculus and first-order logic. Our approach enables categorization of tempo-modal propositions under a time ontology, structured according to a formalism that we previously introduced. In our time ontology, we identify through syntactic proof 32 n-ary subsumption relations among the 33 temporal concept types, forming a hierarchy that could be graphically represented as a tree structure. Some axioms and properties linking our temporal logic with propositional calculus are also identified, contributing to future research in combining time and event ontologies. Possible world semantics and multi-agent systems are other directions that could be explored in the future in conjunction with our concept of subjectivity in time and event. Our ultimate aim is to use our temporal logic to assist formal reasoning involving time, including the

development of the Semantic Web, e.g., by describing the temporal content of web pages and by building automated natural language translation engines.

8 References

- Allen, J.F. (1984): Towards a general theory of action and time. In *Artificial Intelligence*, Vol. 23.
- Bittner, T. (2002): Approximate Qualitative Temporal Reasoning. In *Annals of Mathematics and Artificial Intelligence*, 36(1-2):39-80.
- Corbett, D. (2003): *Reasoning and Unification over Conceptual Graphs*, New York, Kluwer Academic Publishers.
- Klement, K.C. (2006): Propositional Logic. In *Internet Encyclopedia of Philosophy*, James Fieser and Bradley Dowden (eds.), University of Tennessee, USA.
- Hobbs, J.R. and Pan, F. (2004): An Ontology of Time for the Semantic Web. In *ACM Transactions on Asian Language Processing (TALIP): Special issue on Temporal Information Processing*, ACM Inc.
- Hobbs, J.R. (2004): An OWL Ontology of Time. Available at: <http://www.isi.edu/~pan/time/owl-time-july04.txt>
- Lejewski, C. (1959): Time and Modality by A. N. Prior. In *Philosophy*, 34:56-59.
- McTaggart, J.E. (1908): The Unreality of Time. In *Mind: A Quarterly Review of Psychology and Philosophy*, 17:456-473.
- Nguyen, P. and Corbett, D. (2006): A Basic Mathematical Framework for Conceptual Graphs. In *IEEE Transactions on Knowledge and Data Engineering*, 18(2):261-271.
- Nguyen, P. and Corbett, D. (2006): Building Corporate Knowledge through Ontology Integration. *Pacific Rim Knowledge Acquisition Workshop (PKAW06)*, Guilin, China. LNAI 4303.
- Øhrstrøm, P. and Schärfe, H. (2004): A Priorean Approach to Time Ontologies. *12th Int. Conf. on Conceptual Structures (ICCS04)*, Huntsville, USA. LNAI 3127.
- Smith, P. (2003): *An Introduction to Formal Logic*, Cambridge University Press.
- Sowa, J. (1984): *Conceptual Structures - Information Processing in Mind and Machine*, Addison-Wesley.
- Stein, L. and Morgenstern, L. (1994): Motivated Action Theory - A Formal Theory of Causal Reasoning. In *Artificial Intelligence*, 71:1-42.
- Zhou, Q. and Fikes, R. (2000): A Reusable Time Ontology. Knowledge Systems Laboratory. Stanford University. Available at: ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-00-01.htm.gz

Dealing with the Formal Analysis of Information Security Policies through Ontologies: A Case Study

G. M. H. da Silva¹ A. Rademaker¹ D. R. Vasconcelos² F. N. Amaral³ C. Bazílio³
 V. G. Costa¹ E. H. Haeusler¹

¹ TECMF – Departamento de Informática
 Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio,
 Email: {hamazaki, arademaker, vaston, hermann}@inf.puc-rio.br

² Departamento de Computação
 Universidade Federal do Ceará – UFC
 Email: {daviromero}@ufc.br

³ Departamento de Ciência e Tecnologia
 Pólo Universitário de Rio das Ostras, Universidade Federal Fluminense – PURO – UFF
 Email: {fnaufel, bazilio}@ic.uff.br

Abstract

We present the structure of an ontology for Information Security (IS), applied to the extraction of knowledge from Natural Language texts (IS standards, security policies and security control descriptions). This ontology is composed of the vocabulary for the IS Domain, and a particular kind of ontology description, logical forms to determine the structure of the DL formulas associated with the texts. We also discuss the relationship between the structure of the formulas and the efficiency of the reasoner.

Keywords: Information Security Policies, Description Logic, Ontologies

1 Introduction

Information Security (IS) is a non-trivial problem that usually comes in different levels of abstraction. At the *Business* level of abstraction we can find the information related to *Processes* and *Persons* involved in any of the enterprise's activities. At the *System* level we find the *Software* and *Hardware* under safeguard. The *Enterprise's* or *Organization's* security management recognizes (Caralli 2004) this problem as an important question in its own right, not merely as a technological problem to be solved by means of software installation (as *Firewalls*, for example). Social aspects should be also taken into account¹ when studying security problems at the *Process* or *Person* level. On the other hand, there are many attacks reported at the *System* level², enough to take the technological problem also into account. Needless to say, a broken security protocol can prevent an *organization* from fulfilling its social role.

In order to solve the IS problem, two main approaches have been adopted: one based on the defense against predicted (or rather, hopefully predicted) threats, and the other based on maintaining the organization at previously established security levels. The former, named *Threat-Based IS* approach, tries to mount a strong defense against

likely attacks, while the latter maintains the behavior of each entity in the organization at ever-controlled states. Most of the *security standards*, as well as *security protocols*, seem to adhere to each of the two approaches at some level, not necessarily to the exclusion of the other.

The IS community has created sets of *rules*, in a quite artificial variety of Natural Language, in the form of security conditions to be verified. These rules have been continuously updated, and an organization must satisfy them in order to be considered *secure*. Well-known examples of these sets of rules are the *standards* provided by some committees (ISO, COSO, ANSI, Brazil's ABNT, etc). Each set of rules, designed to a specific level of abstraction, carries a *terminology*, which is basically formed by the *linguistic* terms that denote the concepts involved in the specific domain of IS. One can verify by reading a typical *standard*, for example ISO-27001 (ISO 2005b), that it is presented as a set of phrases in a quite artificial Natural Language pattern. The reason for that is the (intended) lack of ambiguity that such texts must have. Besides the *standards*, each organization has its IS Policy (**ISP**), whose level of abstraction is significantly lower than that of a *standard*. One can easily verify that a condition in a *standard* must be turned into an obligation at the **ISP** level. And of course, in order to be implemented, this obligation must be written in a way that will facilitate the task of verifying the status of its own implementation. Thus, in *IS* terms we have two sets of rules and we should ensure that each rule of the higher-level set is covered by the rules of the lower-level set in every viewpoint for the application of the former. We call this *compliance testing*. Thus, besides the task of designing rules and proving them to be adequate to "reality" (the organization and its surroundings), there is the task of comparing, by means of some form of compliance testing, two or more sets of rules. This is what we call the **formal** statement of the IS problem.

For the people in the Ontology community, the above pictured scenario makes for an invitation to work. Ontologies are formalizations of conceptual worlds, and they also serve to prove certain properties about them. These proofs may show the adequacy of the Ontology to its object. Among those important properties one may cite (1) consistency testing, which ensures that the ontology has a model (i.e., it "speaks" about something), and (2) hierarchy classification (subsumption of concepts), which verifies the partial order among concepts. These tests are quite useful in our formal statement of the IS problem. In fact, there are currently many projects and initiatives worldwide that do just that. The references would be too numerous for us to cite them all. Only as a matter of pref-

Copyright ©2007, Australian Computer Society, Inc. This paper appeared at the 3rd Australasian Ontology Workshop (AOW-07), Gold Coast, Queensland, Australia. Conferences in Research and Practice in Information Technology, Vol. 85. Editors, Thomas Meyer and Abhaya C. Nayak. Reproduction for academic, not-for profit purposes permitted provided this text is included.

⁰Research partially supported by CNPq grant 550652/05-1, The Anubis Project.

¹Social Engineering techniques can be envisaged in order to design an attack to access the Enterprises ordinary trash, by means of an intruder personified as an employee or acting with the help of other employees.

²CERT/CC (CERT/CC 2007) keeps records on most of them.

erence, we point out the repository of ontologies placed by the users of the Protégé Ontology Editor (Stanford University 2006).

In (do Amaral et al. 2006) we describe the **Anubis** project, an architecture and a set of formal tools, as well as a methodology, to help a Security Company in designing, validating and maintaining a knowledge base on **ISPs**. As a case study and with the purpose of preserving industrial property (do Amaral et al. 2006), we omit many details, and in certain moments the actual names and denominations inside the architecture were presented in a more general way. The relevant information is briefly presented in Section 2. The structure used to represent Natural Language constructs in order to facilitate the task of formalizing IS concepts and rules from NL Norms is presented in Section 3. The aim of the present article is to report noteworthy conclusions that have been drawn during the development of the above mentioned project.

We focus on the two main conclusions discussed at the end of this article. Firstly, the particular style of Ontology Description, as a set of DL formulae, has become critical in terms of efficiency. It is interesting to note how the presentation style of “equivalent” sets of DL formulas (“equivalent” by means of a syntactic homomorphism) can produce highly different performance results in validation. Although linguistic features of the text to be formalized could argue in favor of a **nested logical form** representation, the use of **flat** representations has shown better computational performance. This is detailed in Section 4. Secondly, and not surprisingly, many **actions** conceived from a set of controls by a domain specialist have not been successfully validated against their respective sets of controls, at least not without the addition of particular axioms stating either obvious or automatic effects, transparent to the domain specialist. Section 4 lists some examples where this situation happened.

2 A brief presentation of the case study

The formalization of text-based information is an important issue in the deployment of semantics-aware technologies in the enterprise. It is very common to encounter situations where knowledge stored in natural-language documents must be made available to agents (human or software-based) for processing and decision-making. This case study can be seen as an attempt to provide an ontology-based approach to the formalization of normative texts in the domain of Information Security (IS), such as security policies defined by organizations and *standards* defined by Security Committees. In (do Amaral et al. 2006) we discuss the principles involved in the development of this approach.

Because the IS-related terminology tends to vary according to the source, we adopt the following definitions: a *standard* is a public document consisting of a set of *control objectives*, which are goals to be attained by the organization if a great level of security is desired. Roughly speaking, control objectives state *what* should be achieved; being expressed at a rather high level of abstraction, they do not lend themselves to direct application to the organization’s processes and practices. It is by means of *security controls* that the organization actually specifies *how* to achieve the security requirements laid out by the control objectives. Security controls (or simply *controls*) are low-level technical measures that can be deployed in order to protect the organization’s devices and processes against potential threats. To bridge the gap between high-level control objectives and low-level controls, the organization defines its *security policy*, consisting of *actions* to be taken in order to comply with the adopted *standards* and possibly with other security requirements identified by a process of risk analysis. In this scenario, one control objective may give rise to several different actions in the

security policy, and each of those actions may be implemented by a set of different controls.

Many tasks are involved in the formalization of the IS domain as described above: for example, *standards* must be selected, actions must be formulated, controls must be defined, deployed and managed. Furthermore, all levels must support maintenance: updates in the *standards* must be followed, policies must be revised, and controls must be replaced or incremented because they become ineffective, inapplicable or simply insufficient. It should be clear that security experts can greatly benefit from the use of semi-automatic, knowledge-based and formal tools to assist them in these activities. From the computer science community viewpoint, we would say that we focus on the use of CAV³ tools. In fact, in our case study we already have a set of *controls*, which comprise the knowledge base of a security analysis system marketed by our industrial partner. By an abuse of language, we call this set of *controls* the IS Knowledge Base (**ISKB**). Thus, the approach to be followed is to group *controls* into *actions*, checking their respective consistency and verifying the compliance of the group of *controls* with regard to the respective *actions*. Subsequently, *actions* and *control objectives* should be checked for compliance too.

Before explaining our approach in more detail, it would be interesting to mention the almost natural boundary conditions of our industrial scenario: (1) *Control objectives* cannot be modified neither in their contents nor in their form, since they are rigid documents (*the standards, for example NIST, COBIT, etc.*); (2) *Actions* are designed by the human being for better clustering and understanding of the Base of *controls* ISKB; actions can and must be modified as a way to easily reach an understandable compliance between the *KB* and the chosen *standard*; (3) Modification of *Controls* (*Security Controls*) should be avoided, and their level of abstraction should be the lowest possible.

Our approach consists of the following elements:

- *Actions* are represented at the *logical form* level, a concept from the area of natural language understanding (Allen 1995). Basically, logical forms are constructs in some suitable formalism used to represent the context-independent semantics of natural language utterances. Currently the edition of *actions* and *controls* (and *control objectives* in the future) is accomplished by means of a Protégé Plug-in developed as an Ontology-Driven editor guided by the *logical forms* ontology.
- The inference capabilities of the proposed framework are based on a description logic (DL) (Baader et al. 2003). Logical forms representing actions are actually stored as DL concepts, and the facts that must hold about these concepts are stored as DL axioms. The user may pose queries to a DL reasoner, which will provide answers based on these axioms. Some background on DL is assumed in this article. The reasoner used in the experiments here reported is the Pellet implementation of the tableaux proof method for DL. Note that the IS domain and the need for additional axioms as reported in Section 4 could require a more powerful DL, but this has not happened in practice.
- Ontologies (Gruber 1993) serve as the unifying structure for the above two elements. An ontology consists of concepts, properties and logical expressions denoting constraints that hold between these concepts and properties. In our approach, actions in logical forms and axioms about them are expressed in terms of concepts, properties and constraints. One language for representing ontologies is OWL DL (Dean & Schreiber 2004), which can be translated in a straightforward way to the language

³Computer Aided Validation.

used by DL reasoners, providing an easy interface between the ontology and the inference services of our framework.

The goal architecture to help formalize IS from NL texts guided by a domain specialist is depicted in Figure 1. The life-cycle of our IS ontology development is shown in Figure 2.

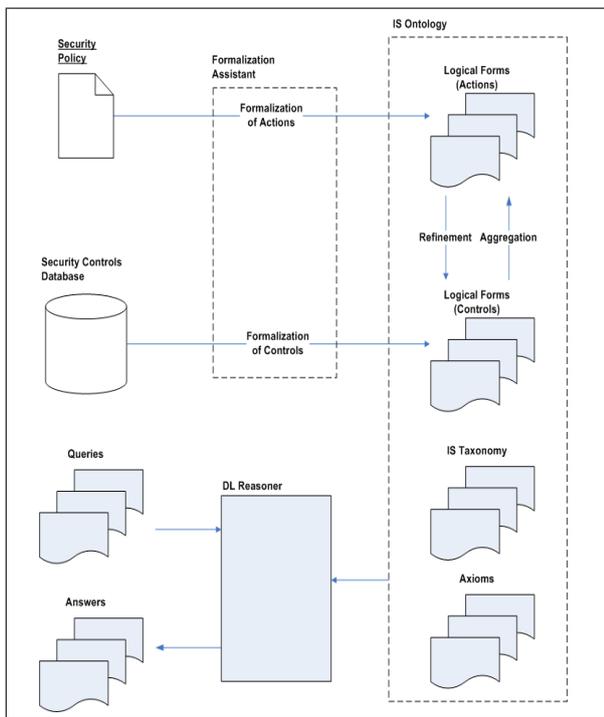


Figure 1: Elements of our ontology-based approach

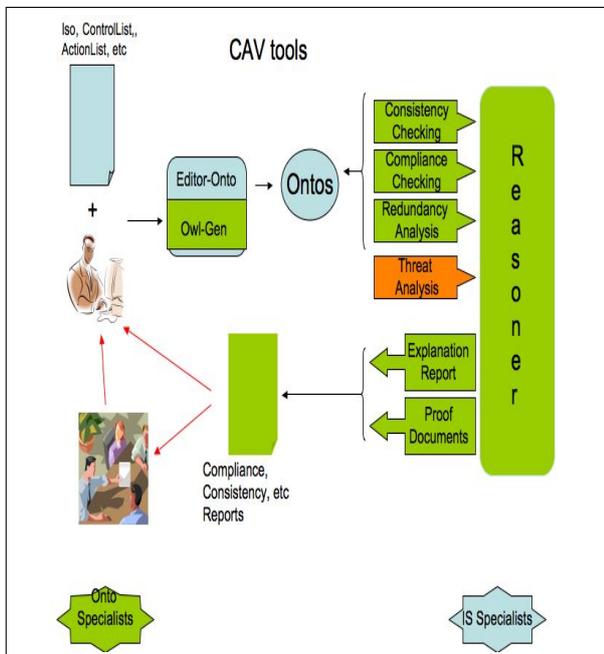


Figure 2: Life-cycle of IS ontology development.

3 The rationale on the IS Ontology defined in the project

In (do Amaral et al. 2006) we discuss the decision to use a structured representation of NL constructs in order to facilitate the task of formalizing IS concepts and rules from NL standards. In fact, almost every rule has a verb (only one verb)⁴ as its most important word, and other elements are linked to it by attributes that are usually (but not always) determined by the syntactic roles they play in the sentence. For example, **hasTheme** is normally associated to the direct object, **hasAgent** to the subject, **hasPurpose** to a subordinated phrase and so on. These structures are described using DL formulas as part of the *Logical forms*, envisaged just in order to solve this situation in a NL processing scenario. We use these as the core of the Ontology. Thus, our IS ontology has a linguistic core that is responsible for describing concepts and relationships essential in structuring Normative texts as an indexed set of sentences (the *actions*, *controls* and *control objectives*). This core is semantically neutral. In order to reason about IS concepts, there must be an Ontology of the IS vocabulary. For example, *Server*, *Firewall*, *Operating System*, *CEO*, *Meeting Room* and so on are concepts belonging to this IS vocabulary.

One of the points that were strongly stressed during the project was the construction of the ontology by the domain specialist himself. The IS specialist is used to working with Natural Language texts, namely, *standards*, laws, company policies, security protocols (not only software, but also human-based security protocols) and so on. It was noted that the specialist prefers to keep working this way: his focus is on the written material. Discussing the general extraction of knowledge from Natural Language texts is outside the scope of this project; however, there is clearly a need for some methodology and a tool to support it. In order to solve this, a Protégé plug-in was implemented. This plug-in is an editor guided by the core ontology (an OWL-DL document), which, by interacting with an IS specialist, produces a set of DL-formulae representing either an IS vocabulary, or an *action*, or a *control*, or a *control objective*. The editing process works by marking and typing with DL meaning the linguistic elements of a normative text. In fact, the plug-in was not used for the edition of the almost 2000 controls and actions that make up the last version of the Ontology. Besides that, part of the vocabulary related to IS on the *Linux* Operating System was defined by an IS specialist (an employee of the Industrial Partner).

The main advantage of the use of *logical forms* is its compliance with the guidelines for Norms constructions (see (ISO 2005a)). For example, in most cases our ontology does not, and should not, distinguish between different parts of speech, so as to render the constructed logical forms as general as possible; thus, a noun whose root is shared with a verb is represented by the concept associated to the verb; for example, “connect” and “connection” are both associated to the single concept Connect, and “configure” and “configuration” are both associated to the single concept Configure. In (ISO 2005a), Sect. 6.4 states that all concepts should be preferentially stored in noun form; we have chosen verb form instead of noun form because the actions in a security policy have verbs as their most important words.

Before we describe some examples that have allowed us to draw the conclusions mentioned in sections 1 and 5, a few words about our IS ontology (from the logical point of view) are in order. The main feature of our IS ontology is that it is a T-Box ontology. There are no individuals asserted, nor is there any mention of them. This is justified by observing that from the point of view of IS, individuals seem to have no importance. For example, a *Person* invading a private area of a corporation equals

⁴Modal forms must be distilled, since they are in general redundant.

any other invader in the same conditions. It is not important to name the invader, but to describe him or her. The same can be said about the facilities of the corporation (organization), about the CEO, the *Operating System*. A password instance is unimportant whenever comparing the two concepts *valid password* and *invalid password*. Besides, because we deal with negation, disjunctions and conjunctions, and the core ontology uses restrictions, we use *ALC* (Baader et al. 2003) as our logic language. This has the (desirable) side-effect of not going so high in the complexity hierarchy of DLs. Basically we are inside PSPACE-complete worst-case complexity. This is not so far⁵ from the usual reasoning complexity for knowledge representation systems.

The next section is the core of this article, showing some interesting examples that appear in the IS ontology under construction. Both conclusions mentioned in the introduction have been drawn from almost the same kind of formalization. Thus, both claims are analyzed in the sequel.

4 Formalization of Controls and Actions of IS

In this section we illustrate the formalization of IS actions and controls from normative texts. We consider a few examples of actions and controls formalized by nested and flat representations. These examples provide samples of the different types of axioms that are needed in the ISKB in order to accomplish compliance tests. It is worth mentioning that the automatic process is simpler in flat form (in which the axioms are more modular) than in the nested style. In addition, the validation is also more efficient using flat forms than nested forms. At the end of this section, we show the performance of both approaches.

Before going into the examples, a short explanation. The *nested* style of specifying *Logical Forms (LF)* was our first choice; it was induced by the nested style of the modifiers (modalities, adverbs, subordinated phrases, etc.) usually found in Natural Language sentences. The use of attributes (roles) in our *LF* linguistic ontology as a way of specifying the role of each phrasal element in the phrase is a natural one. We recall that if R_1 and R_2 are DL roles, and C and D are concepts, there is no logical equivalence between the concepts $\exists R_1.(C \sqcap D)$ and $\exists R_1.C \sqcap \exists R_1.D$. Analogously, for general R_1 and R_2 , $\exists R_1 \exists R_2.C$ and $\exists R_2 \exists R_1.C$ are not logically equivalent concepts either. Note that in this last case we can say that $\exists R_1.C$ is in the context of R_2 and not the other way around. We have an explicit dependence of R_1 on R_2 .

Thus, by taking the *nested* way of specifying the *LF* ontology, we force the dependence everywhere. This is not bad, if this is done on the whole set of *actions* and *controls* consistently. This has the advantage of generating a better explanation of how a *control* is subsumed by a particular *action*. Another advantage of the nested style is the fact that it provides a more automatic way of rendering NL sentences into *LF*, for the sequence of modifiers is already in the sentence itself. However, as anyone can observe, the use of passive voice, indirect styles of speech and similar features that Natural Languages display, might force us to consider the advantages of the *nested* style as *apparent* advantages. In fact, the first experiments conducted with our knowledge base (KB) were based on the use of the *nested* style. This style often allowed one to visually check subsumption of a *control* by an *action* by simply placing both descriptions side by side. Such descriptions were, in most cases, of the same form when considering the replacements induced by the *IS* ontology. For example, from the action: "Enable the directive "use-id-pool" in DNS server", the control: "The directive "use-id-pool" should be used in the DNS Bind server", $DNSBindServer \sqsubseteq DNSServer$ and $Enable \sqsubseteq Use$,

⁵We believe $NP \neq PSPACE$.

we have:

$$\begin{aligned} & \exists hasVerb.(Use \sqcap \exists hasTheme.DirectiveUseIdPool \\ & \quad \sqcap \exists hasLocation.DNSBindServer) \\ & \quad \sqsubseteq \\ & \exists hasVerb.(Enable \sqcap \exists hasTheme.DirectiveUseIdPool \\ & \quad \sqcap \exists hasLocation.DNSServer) \end{aligned}$$

Example 4.1. Suppose the organization has deployed an action with the following description:

Configure the "type" parameter with the value "Nt5DS" in the [Windows OS].

The nested and flat logical form expressions that represent such action are the concepts defined by:

$$\begin{aligned} ActionINested & \equiv \\ 1 & \exists hasVerb.(Configure \sqcap \\ 2 & \quad \exists hasTheme.(TypeParameter \sqcap \\ 3 & \quad \quad \exists hasValue.(Nt5DS)) \sqcap \\ 4 & \quad \exists hasLocation.Windows) \end{aligned}$$

$$\begin{aligned} ActionIFlat & \equiv \\ 1 & \exists hasVerb.Configure \sqcap \\ 2 & \quad \exists hasTheme.(TypeParameter \sqcap \\ 3 & \quad \quad \exists hasValue.(Nt5DS)) \sqcap \\ 4 & \quad \exists hasLocation.Windows \end{aligned}$$

Now consider the organization has deployed a security control with the following description:

The Registry "type" parameter must be configured with the value "Nt5DS".

The nested and flat logical form expressions that represent this control are the concepts defined by:

$$\begin{aligned} ControlINested & \equiv \\ 1 & \exists hasVerb.(Configure \sqcap \\ 2 & \quad \exists hasTheme.(TypeParameter \sqcap \\ 3 & \quad \quad \exists hasValue.(Nt5DS \sqcap \\ 4 & \quad \quad \quad \exists hasPossessor.WindowsRegistry))) \end{aligned}$$

$$\begin{aligned} ControlIFlat & \equiv \\ 1 & \exists hasVerb.Configure \sqcap \\ 2 & \quad \exists hasTheme.(TypeParameter \sqcap \\ 3 & \quad \quad \exists hasValue.(Nt5DS)) \sqcap \\ 4 & \quad \exists hasPossessor.WindowsRegistry \end{aligned}$$

To prove the compliance of our structures, we need to add axioms that correlate the location of the Windows parameter with the Windows Registry. As we already said, the nested approach is more complicated than flat approach even in the used axioms.

The following axioms are necessary for the nested formalization.

$$WindowsTypeParam \equiv RegWindowsTypeParam$$

$$\begin{aligned} WindowsTypeParam & \equiv \\ 1 & \exists hasTheme.(TypeParameter \sqcap \\ 2 & \quad \exists hasValue.(Nt5DS)) \sqcap \\ 3 & \quad \exists hasLocation.Windows \end{aligned}$$

$$\begin{aligned} RegWindowsTypeParam & \equiv \\ 1 & \exists hasTheme.(TypeParameter \sqcap \\ 2 & \quad \exists hasValue.(Nt5DS \sqcap \\ 3 & \quad \quad \exists hasPossessor.WindowsRegistry)) \end{aligned}$$

The following axioms are necessary for the flat formalization.

$$\begin{aligned} \text{WindowsLocation} &\equiv \text{WindowsRegPossessor} \\ \text{WindowsLocation} &\equiv \exists \text{hasLocation. Windows} \\ \text{WindowsRegPossessor} &\equiv \\ &\exists \text{hasPossessor. WindowsRegistry} \end{aligned}$$

In the following example, we illustrate that some representations do not need additional axioms.

Example 4.2. Suppose the following action is part of the organization's security police:

Set the Date and Time of [system xyz].

The nested and flat logical form expressions that represent such action are the concepts defined by:

$$\begin{aligned} \text{Action2-Nested} &\equiv \\ 1 \quad &\exists \text{hasVerb. (Set } \sqcap \\ 2 \quad &\exists \text{hasTheme. (DateTime } \sqcap \\ 3 \quad &\exists \text{hasLocation. (System)))} \end{aligned}$$

$$\begin{aligned} \text{Action2-Flat} &\equiv \\ 1 \quad &\exists \text{hasVerb. Set } \sqcap \\ 2 \quad &\exists \text{hasTheme. DateTime } \sqcap \\ 3 \quad &\exists \text{hasLocation. System} \end{aligned}$$

Suppose the organization has deployed a security control with the following description:

The date and time of Linux Red Hat must be configured.

The nested and flat logical form expressions that represent such control are the concepts defined by:

$$\begin{aligned} \text{Control2-Nested} &\equiv \\ 1 \quad &\exists \text{hasVerb. (Configure } \sqcap \\ 2 \quad &\exists \text{hasTheme. (DateTime } \sqcap \\ 3 \quad &\exists \text{hasLocation. (LinuxRedHat))} \end{aligned}$$

$$\begin{aligned} \text{Control2-Flat} &\equiv \\ 1 \quad &\exists \text{hasVerb. Configure } \sqcap \\ 2 \quad &\exists \text{hasTheme. DateTime } \sqcap \\ 3 \quad &\exists \text{hasLocation. LinuxRedHat} \end{aligned}$$

In this example, there was no need for additional axioms to prove compliance of this control with this action, since we have $\text{Set} \equiv \text{Configure}$ and $\text{LinuxRedHat} \sqsubseteq \text{OperatingSystem} \sqsubseteq \text{System}$ in the ISKB.

The next example illustrates the need for a more sophisticated set of axioms in order to show that an action subsumes a control or a set of controls.

Example 4.3. Suppose the following action is part of the organization's security policy:

Define TCP port with a non-default value for the execution of [xyz service].

The nested and flat logical form expressions that represent this action are the concepts defined by:

$$\begin{aligned} \text{Action3-Nested} &\equiv \\ 1 \quad &\exists \text{hasVerb. (Define } \sqcap \\ 2 \quad &\exists \text{hasTheme. TCPPort } \sqcap \\ 3 \quad &\exists \text{hasValue. (TCPNonDefaultValue) } \sqcap \\ 4 \quad &\exists \text{hasPurpose. (Execute } \sqcap \\ 5 \quad &\exists \text{hasTheme. SoftwareService)} \end{aligned}$$

$$\begin{aligned} \text{Action3-Flat} &\equiv \\ 1 \quad &\exists \text{hasVerb. Define } \sqcap \\ 2 \quad &\exists \text{hasTheme. TCPPort } \sqcap \\ 3 \quad &\exists \text{hasValue. TCPNonDefaultValue } \sqcap \\ 4 \quad &\exists \text{hasPurpose. (Execute } \sqcap \\ 5 \quad &\exists \text{hasTheme. SoftwareService)} \end{aligned}$$

Now suppose the organization has deployed a security control with the following description:

Apache Tomcat must be executed in a non-default port and in a port not allocated to reserved services.

The nested and flat logical form expressions that represent such security control are the concepts defined by:

$$\begin{aligned} \text{Control3-Nested} &\equiv \\ 1 \quad &\exists \text{hasVerb. (Execute } \sqcap \\ 2 \quad &\exists \text{hasTheme. (ApacheTomcat } \sqcap \\ 3 \quad &\exists \text{hasLocation. (TCPPort } \sqcap \\ 4 \quad &\exists \text{hasValue. (TCPNotReservedServicePort) } \sqcap \\ 5 \quad &\exists \text{hasValue. (TCPNonDefaultValue))} \end{aligned}$$

$$\begin{aligned} \text{Control3-Flat} &\equiv \\ 1 \quad &\exists \text{hasVerb. Execute } \sqcap \\ 2 \quad &\exists \text{hasTheme. ApacheTomcat } \sqcap \\ 3 \quad &\exists \text{hasLocation. TCPPort } \sqcap \\ 4 \quad &\exists \text{hasValue. (TCPNotReservedServicePort } \sqcap \\ 5 \quad &\exists \text{hasValue. TCPNonDefaultValue)} \end{aligned}$$

In any case, the structure of the control is quite different from the structure of the action. In fact, the abstraction level of the action seems to be higher. However, taking into account the meaning of both sentences, we can see that the subsumption would hold if we established that "To define some X in a way Z in order to execute some Y " is subsumed by "To execute some Y in a way Z (different from the default) running in location X ". In our specific case, this is expressed by the following axioms:

$$\begin{aligned} \text{RightHand-on-Defining} &\equiv \\ 1 \quad &\exists \text{hasVerb. Define } \sqcap \\ 2 \quad &\exists \text{hasTheme. TCPPort } \sqcap \\ 3 \quad &\exists \text{hasValue. TCPNonDefaultValue } \sqcap \\ 4 \quad &\exists \text{hasPurpose. (Execute } \sqcap \\ 5 \quad &\exists \text{hasTheme. SoftwareService)} \end{aligned}$$

$$\begin{aligned} \text{LeftHand-on-Defining} &\equiv \\ 1 \quad &\exists \text{hasVerb. Execute } \sqcap \\ 2 \quad &\exists \text{hasTheme. SoftwareService } \sqcap \\ 3 \quad &\exists \text{hasLocation. (TCPPort } \sqcap \\ 4 \quad &\exists \text{hasValue. (TCPNotReservedServicePort } \sqcap \\ 5 \quad &\exists \text{hasValue. TCPNonDefaultValue)} \end{aligned}$$

$\text{LeftHand-on-Defining} \sqsubseteq \text{RightHand-on-Defining}$

The above axiom, together with the following axiom belonging to the ISKB, finally prove the required compliance between the control (flat) and the action (flat). For the nested version the axioms are analogous.

$\text{ApacheTomCat} \sqsubseteq \text{SoftwareService}$

The last thing to note in this section is the difference of performance between the two styles of formalization, nested and flat. In order to compare performances, a small part of the ISKB was extracted. The classified ontology has 46 controls, 8 actions and 10 (additional) axioms. The time spent by Pellet was 22.844 seconds, the nested case, and, 4.734 seconds, the flat case. These experiments were

executed on the same machine, a Pentium IV with 1 GB RAM, under similar conditions (no other application was running together with **Pellet**, but **Protege** itself).

5 Conclusion

In section 4, we have shown how two stylistically different but homomorphic⁶ DL representations for *logical forms (LF)* can exhibit quite different performances when used for automatic verification of compliance. The initial set of *LF* attributes had 27 different roles, and it was designed with the explicit purpose of covering almost linguistic aspect of a phrasal utterance in the IS domain. But at the beginning of the studies, it was realized that a smaller set would be enough. A subset of the initial set, containing only 8 *LF* attributes, was defined. The *actions* that are shown in this article already use only 8 *LF* attributes. The current stage of the IS ontology has been tuned according to the main observations on under-specification (lack of essential formalization, as for example the axioms shown in section 4), and according to performance.

There are others tools that extract the knowledge from IS domains texts, for example (Bonatti et al. 2004), but the main advantage of our approach is the usage of logical forms to obtain the compliance testing.

During the development of this case study, besides the above mentioned conclusions, a methodological conclusion concerning the interaction with the specialist is worth noting: the tools must be designed to serve the needs of those who have asked for the tool. In this sense, we plan to add to our architecture and set of tools a proof explanation generator to allow the specialist to analyze the validation at almost the same level of abstraction as the specification (i.e., the ontology).

Last, but not least, it is important to note that the overall approach shown here can, in principle, also be applied to any other domain with similar features, especially with a body of knowledge written in adequate NL form.

References

- Allen, J. (1995), *Natural Language Understanding*, 2nd edn, Benjamin Cummings.
- Baader, F., Calvanese, D., McGuinness, D., Nardi, D. & Patel-Schneider, P., eds (2003), *The Description Logic Handbook*, Cambridge University Press.
- Bonatti, P., Shahmehri, N., Duma, C., Olmedilla, D., Njdl, W., Baldoni, M., Baroglio, C., Martelli, A., Patti, V., Coraggio, P. et al. (2004), Rule-based policy specification: State of the art and future work, Technical Report I2: D1, Rewerse, Reasoning on the web. Available at <http://rewise.net/deliverables/i2-d1.pdf>.
- Caralli, R. A. (2004), 'Managing for enterprise security', Technical Note CMU/SEI-2004-TN-046. Available at http://www.sei.cmu.edu/publications/documents/04_reports/04tn046.html.
- CERT/CC (2007), 'Incident notes'. http://www.cert.org/incident_notes/.
- Dean, M. & Schreiber, G. (2004), 'OWL Web Ontology Language Reference', <http://www.w3.org/TR/owl-ref/>.
- do Amaral, F. N., Bazílio, C., da Silva, G. M. H., Rade-maker, A. & Haeusler, E. H. (2006), An ontology-based approach to the formalization of information security policies, in 'VORTE - Workshop on Vocabularies, Ontologies, and Rules for the Enterprise', Hong Kong.
- Gruber, T. R. (1993), Towards principles for the design of ontologies used for knowledge sharing, in N. Guarino & R. Poli, eds, 'Formal Ontology in Conceptual Analysis and Knowledge Representation', Kluwer Academic Publishers.
- ISO (2005a), *ANSI/NISO Z39.19-2005, Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies*, NISO.
- ISO (2005b), *BS ISO/IEC 27001 Stand Alone*, ISO/IEC.
- Stanford University (2006), 'The Protégé Ontology Editor and Knowledge Acquisition System', <http://protege.stanford.edu>.

⁶This means that there is a function mapping one style into the other which preserves the subsumption relationship.

Extraction, Evaluation and Integration of Lexical-Semantic Relations for the Automated Construction of a Lexical Ontology

Tonio Wandmacher, Ekaterina Ovchinnikova, Ulf Krumnack, Henrik Dittmann

Artificial Intelligence group
 Institut für Kognitionswissenschaft
 Universität Osnabrück, Germany
 Email: {firstname.lastname}@uni-osnabrueck.de

Abstract

Several approaches for extracting semantic relations from various types of resources have been proposed during the last years. While already of great value when used separately, combining these techniques promises to lead to even broader and more reliable results. However, divergent information may occur when assembling such data. We present LEXO, a framework for integrating semantic relations from different sources into an ontological structure. We provide different methods for assigning confidence values to the input data as well as mechanisms to detect and resolve inconsistencies. The present paper focuses on lexical-semantic relations, but the approach presented is extensible to include new kinds of data sources as well as further types of relations.

1 Introduction

The construction of ontologies is considered essential not only in the development of the semantic web but also for a growing number of natural language processing (NLP) tasks such as word sense disambiguation, automatic semantic annotation of documents, question answering, machine translation and anaphora resolution.

Whereas most ontologies are constructed for a given domain and contain relations between concepts, a *lexical ontology* is intended to provide structured information on words of a given language and their semantic relatedness; meaning is encoded by relating a given lexical item to others. Also, the main goal of a lexical ontology is not to store general encyclopedic or ontological knowledge, but to serve as common database, assembling lexical and semantic information.

In the past years a number of projects have been presented that try to achieve this goal, of which the most prominent one is the Princeton WordNet (Fellbaum 1998). It represents domain independent, lexical-semantic knowledge in a network-like structure which makes taxonomic relationships explicit. However, it cannot be considered as an ontology in the formal sense, since the relations are based on linguistic evidence rather than on formal ontological principles, and it does not guarantee any kind of consistency (cf. (Oltamari et al. 2002) for examples of ontological inconsistencies in WordNet).

The main problem, however, remains data coverage. Even though WordNet and its cousins are con-

sidered as broad coverage resources, many NLP applications run into problems of data sparsity when relying on such resources only, which are all developed manually at great cost. A possible solution to the sparsity problem present automatic extraction procedures. In the past years a lot of automated approaches have been presented to extract ontological knowledge from text or even structured data (for an overview cf. Maedche 2002 or Cimiano 2006). The main problem of these approaches however is their reliability; as every unsupervised procedure, they also extract noise. A way to overcome the problems of low coverage and low data quality is the cumulation of evidence. When many available resources and extraction procedures are exploited at the same time, reliable relations can be distinguished from noise, if practical measures to estimate the confidence of each relation are provided.

To our knowledge however, no bigger attempt has been made to realize this idea. The approach described by Cimiano et al. (2005) has gone in this direction by integrating taxonomic relations from different ontology learning paradigms. Another interesting work (Snow et al. 2006) presents an algorithm to induce a domain-independent taxonomy from heterogeneous resources by defining several constraints on the resulting structure. However these approaches only consider *is-a* relations, and they have only been applied on a small scale.

The LEXO project that we present here, aims at integrating any kind of lexical-semantic relation from automated extraction procedures and already existing, freely accessible lexical resources. Information from various origins is cumulated and integrated in a way which makes it possible to identify reliable relations. These relations will form a set of *hypotheses* from which an ontology is constructed. Our approach is highly automated. We present an elaborate measure to estimate the confidence for each incoming relation hypothesis. Our confidence measure takes into account the *a priori* confidence of the respective resource, semantic similarity between the connected terms and structural evidence from the already existing data. The ontology construction itself is automated as well, we define structural consistency conditions which have to be assured by the ontology to be constructed from the assembled relation hypotheses.

Although our work is focused on the creation of a lexical ontology for the German language, the overall approach is in principle language neutral: Methods to extract semantic relations have of course to be designed for an individual language, but they can easily be adapted to other languages. There might also exist other lexical-semantic resources to exploit; our framework takes advantage of any lexical resource and extraction method, as long as it can provide binary relations. Moreover, the types of relation are not fixed either; every relation can be modeled as long as

a resource is able to provide it.

At present, LEXO comprises 975,570 relation entries (synonymy, hyponymy, meronymy and antonymy) over 121,593 unique words (types). So far, we make use of the following resources: *Wiktionary*, *OpenThesaurus* (Naber 2005), *Projekt Deutscher Wortschatz*¹, an (unsupervised) translation of WordNet and an automatic extraction method, looking for lexico-syntactic patterns on the web (similar to Cimiano & Staab 2004).

Since the LEXO project is in an early stage of development, we cannot present an overall evaluation of our methods and the hereby constructed ontology yet. The aim of this paper is to present measures to evaluate the confidence of automatically extracted lexical-semantic relations and to describe a way to integrate these relations in a consistent manner.

The paper is structured as follows: In section 2 we give an overview on methods and resources providing lexical-semantic relations, we then (section 3) describe measures to estimate the confidence of these relations, in section 4 we deal with the problem of word senses and formulate consistency conditions for the resulting ontological structure, and in section 5 we present the overall architecture of the LEXO system and describe possible evaluation scenarios. In the final section we then discuss open issues and describe the following steps of our work.

2 Obtaining semantic relations

Semantic relations between some items are relations between meanings of this items; lexical-semantic relations are thus relations between meanings of words (cf. Cruse 1986). The term *lexical ontology* (LO) is rather underspecified in the existing literature. Usually it means that words of a particular language (rather than abstract concepts) are formally defined and connected with each other by lexical-semantic relations such as *synonymy*, *hyponymy* or *meronymy*. WordNet is considered to be the most typical example of LO. In the LexO framework, a lexical ontology is a set of relations over a domain of words or word senses (unlike WordNet, where relations can hold between synsets). Every relation is a set of pairs of objects from the domain.

While LEXO aims at collecting various kinds of relations, this paper focuses on lexical-semantic relations, i.e. relations that are founded on the meaning of words rather than on their form. This section describes different techniques to obtain such relations from various resources.

2.1 Existing approaches in ontology learning

In the past few years a variety of approaches has been presented that aim at extracting conceptual knowledge from unstructured and semi-structured data. These approaches receive a growing importance in the ontology building process, since for many semantic web as well as NLP applications the amount of available knowledge is crucial. Since these methods are unsupervised, their output is usually rather noisy.

So far, most of the approaches are light-weight from a logical point of view; they return logically simple constructions such as concepts, instances, taxonomic relations and other general relations (e.g. *part-of* or *author-of*). Current methods basically make use of three strategies (or combinations of these):

1. *Distributional information*: The co-occurrence of terms within a given context or document is an

important hint for their conceptual relatedness. Moreover, two terms will be similar in meaning if they tend to occur with the same neighbors (2nd order cooccurrence). Different distributional methods (e.g. collocation analysis or *Latent Semantic Analysis*, Deerwester et al. 1990) give a distance measure between two terms that can be used to represent semantic relatedness. Even though this cannot help labeling the type of relation, it gives a reliable clue that can be further used. Clustering techniques for example use this information to form sets of related terms. In hierarchical clustering procedures, these sets of terms are arranged in a hierarchical fashion. The hereby generated cluster hierarchy can be the base for a taxonomical structure, i.e. a hierarchy of concepts. Approaches that use this kind of strategy are for example described by Caraballo (1999) or Cimiano & Staab (2005).

2. *Lexico-syntactic patterns*: The second strategy basically relies on lexico-syntactic patterns, the so-called *Hearst* patterns (Hearst 1992). Here, a text corpus is scanned for characteristic recurring word combinations, typically containing a semantic relation between two terms (e.g. [w_2 , such as w_1] \rightarrow *hyponym*(w_1, w_2)). These approaches however usually suffer from data sparsity, since many word combinations cannot be found in even large corpora. To cope with this fact, efforts been made to harvest these patterns on the web (cf. Brin 1998, Etzioni et al. 2004 or Cimiano & Staab 2004).
3. *Syntactic and morphosyntactic information*: Finally linguistic structures like verb frames and modifier constructions can help extracting conceptual relations. For example, it is easy to infer a hyponymy relation between *car ferry* and *ferry*, since *car* is here a modifier of *ferry* (cf. Buitelaar et al. 2004). Moreover, from the analysis of dependency paths in syntactic derivations, reliable relations can be learned (Katrenko & Adriaans 2006), other methods make use of predicate-argument relations (e.g. Faure & Nédellec 1998). For the extraction of nontaxonomic relations the analysis of selectional preferences of verbs can be very helpful (Wagner 2000).

Techniques based on these strategies can be found in many ontology learning systems, such as *Snowball* (Agichtein & Gravano 2000), *OntoLearn* (Navigli & Velardi 2004), *OntoLT* (Buitelaar, Olejnik & Sintek 2004), and *Text2Onto* (Cimiano & Völker 2005). Most of these systems are concerned with the extraction of the relevant terminology (from which they deduce the respective classes), with the derivation of subsumption relations and with some basic nontaxonomic relations.

2.2 Automatic translation of WordNet

A rich source of relational lexical information are wordnets, especially the English WordNet. WordNet represents knowledge in form of a lexical network. Its organizing units are sets of synonyms (so-called *synsets*), representing word meanings. Two kinds of relations can be distinguished: a) relations connecting individual lexical items and b) relations connecting synsets and thus providing a statement indirectly via the synonymy relation. Both kinds of relations can be used as input for LEXO.

Although nowadays wordnets exist for many languages,² their benefit often is restricted due to lim-

¹<http://wortschatz.uni-leipzig.de/>

²A current list is maintained by the Global WordNet Association at http://www.globalwordnet.org/gwa/wordnet_table.htm

ited size or license issues, like the German GermaNet (Hamp & Feldweg 1997), which is protected. Therefore in our context, a translation of the English WordNet can be a promising alternative. There have been a number of approaches that use bilingual dictionaries to apply automatic and semi-automatic methods to translate WordNet into different languages (e.g. Spanish (Knight & Luk 1994), Japanese (Okumura & Hovy 1994) or Arabic (Khan & Hovy 1997)). Most problems in such approaches are caused by polysemy, mismatches between the bilingual dictionary and WordNet, as well as mismatches in the lexicalization between the languages.

Various techniques have been proposed to deal with ambiguities that arise when mapping dictionary entries to WordNet synsets (cf. Atserias et al. 1997). They are based on additional information from WordNet and the dictionary such as part-of-speech, alternative translations, domain markers, syntactic and semantic annotation or frequency information. Consider for example the synset

{*plant, flora, plant life*}

and a dictionary entry of the form

plant → *Pflanze* [*bot.*]; *Werk*

There are two different translations for *plant*,³ but as *plant* is polysemous in WordNet, it is not clear, which translation should be mapped to the synset. Here a human can use the domain marker [*bot.*] to disambiguate the translation. To make this strategy available for automatic translation methods, WordNet has to be annotated with the domain markers of the dictionary, a feasible task, as there are usually only few domain markers in use.

An alternative strategy to translate synsets with more than one element is to collect the translations for every word in the synset and consider their intersection. In the above example this means to look at the following entries:

flora → *Flora, Pflanzenwelt* [*biol.*]
plant life → *Pflanzenwelt*

Here *Pflanzenwelt* seems to be a promising translation for the synset (this assumption is further strengthened by the fact that *plant life* is monosemous in WordNet). However, in many cases the intersection is empty, but there are translations that are semantically similar. Given a measure for semantic similarity of words in the target language, this can be used in cases when a common translation is missing.

In most cases such disambiguation techniques do not lead to a definitive selection but rather rank the alternatives. In the context of our work, such a ranking can be used to assign a confidence score to induced relation hypotheses.

There is some agreement that an automatic translation will not result in a ready-to-use WordNet for the target language. However, for our approach, relations stemming from such a translation process, annotated with confidence values, are valuable input material. Once an initial lexical ontology is constructed for the target language, it can be used to foster the disambiguation process, providing in turn more confident hypotheses.

2.3 Obtaining relations from electronic dictionaries and thesauri

In recent years, many lexical resources have been made electronically available. A lot of these provide

free access over the internet and often have liberal licenses governing their use and re-distribution. We present three examples for German.

The *Wiktionary* project is an offshoot of *Wikipedia*⁴, the well-known open encyclopedia. Online since 2002, the site provides dictionaries for a large number of languages. Each of these may contain entries from any language, which are explained in the language of the respective dictionary. Like its sister project, *Wiktionary* is a collaborative effort where basically everyone can participate in its construction. Often such a dictionary's base is assembled by automatic extraction from other publicly available sources, however. The German *Wiktionary* has been online since 2004 and currently has 55,000 entries for all languages, of which more than 40% are for German words.

A *Wiktionary* entry for a given word may comprise all kinds of lexical information, such as phonetics, morphological properties, etymology, word senses and semantic relations (e.g., synonyms, antonyms and hypo-/hyponyms). At present, we extract all lexical-semantic relations between German words that can be identified through the page structure and markup, taking note of word senses whenever they are present in the resource.

The project *OpenThesaurus* (Naber 2005) has been online since 2003. A freely accessible and modifiable resource for the German language, *OpenThesaurus* is primarily structured through groups of synonyms. The project aims at organizing these groups in a hierarchical WordNet-like (Fellbaum 1998) manner, starting from a small range of top-level concepts. In doing this, hypo-/hyponym relationships are added between the synonym groups. However, to date only a fraction of groups have been attached to the hierarchy. *OpenThesaurus* provides its data in a variety of formats, such as a plain database dump or a plug-in to *OpenOffice*.

The *Projekt Deutscher Wortschatz*⁵ at the Universität Leipzig is a monolingual German dictionary, comprising more than 9 million full (i.e., inflected) forms and multi word units. The dictionary is largely based on automatic extraction methods for corpora in conjunction with reviewing and editing by human experts and has more restrictive terms of use than the previous examples.

For a given word, information is provided on grammatical status, frequency, topical domain(s) and semantic relations. Example phrases and automatically calculated co-occurrences and collocations are provided as well. This data is available through either a web interface or a number of web services for automated retrieval.

The example in table 1 shows the relations for the noun *Stern* ('star'), as extracted from the resources mentioned above.

3 Calculating confidence

Estimating the reliability of a given relation is a non-trivial problem for an automated approach, but it is crucial to have such a measure in order to build up an ontology of high quality. In the following we show how we calculate our confidence scores, which are comprised of a local confidence value for a given relation as provided by its resource, the overall reliability of its resource, structural criteria and an automatically calculated similarity score. For this purpose we make use of *Latent Semantic Analysis* (LSA), a vector-based method which has been shown to give reliable estimates on semantic similarity.

⁴<http://www.wikipedia.org>

⁵<http://wortschatz.uni-leipzig.de>

³ *Pflanze*: 'botanical plant'; *Werk*: 'factory'/'work'

OpenThesaurus	Wiktionary	Wortschatz Projekt
<i>synonym</i> (Stern ₁ , Asterisk)	<i>hyponym</i> (Stern _a , Himmelskörper)	<i>synonym</i> (Stern, Filmstar)
<i>synonym</i> (Stern ₁ , Asteriskus)	<i>synonym</i> (Stern _a , Gestirn)	<i>synonym</i> (Stern, Gestirn)
<i>synonym</i> (Stern ₁ , Sternchen)	<i>synonym</i> (Stern _a , Fixstern)	<i>synonym</i> (Stern, Star)
<i>hyponym</i> (Stern ₂ , Gestirn)	<i>hyponym</i> (Stern _b , Symbol)	<i>synonym</i> (Stern, Planet)
<i>hyponym</i> (Stern ₂ , Himmelskörper)	<i>synonym</i> (Stern _b , Asterisk)	<i>hyponym</i> (Stern, Gestirn)
<i>synonym</i> (Stern ₂ , Fixstern)	<i>synonym</i> (Stern _b , Sternchen)	<i>hyponym</i> (Stern, Himmelskörper)
<i>synonym</i> (Stern ₃ , Star)	<i>hyponym</i> (Stern _c , Mensch)	<i>hyponym</i> (Stern, Schmuck)
...	<i>hyponym</i> (Stern _c , Kosewort)	...
...

Table 1: Relations for *Stern* ('star') from *Wiktionary*, *OpenThesaurus* and *Wortschatz*.

3.1 LSA-based semantic similarity

Since the early 1990s, Latent Semantic Analysis (LSA) has become a well-known technique in NLP. When it was first presented by Deerwester et al. (1990), it aimed mainly at improving the vector space model in information retrieval, but in the meantime it has become a helpful tool in NLP as well as in cognitive science (cf. Landauer & Dumais 1997). LSA has been shown to give reliable estimates for the semantic similarity between two terms, and it has also been used to enhance automatic hyponymy extraction techniques (Cederberg & Widdows 2003). If two terms receive a high LSA similarity value, they will be somehow semantically related, however LSA cannot determine the kind of relation (Wandmacher 2005).

The LSA model is based on the vector space model from information retrieval (IR) (Salton & McGill 1983). Here, a given corpus of text is first transformed into a term \times context matrix A , displaying the occurrences of each word in each context. Usually, this matrix is then weighted by one of the standard weighting methods used in information retrieval (c.f. Salton & McGill 1983). The decisive step in the LSA process is then a *singular value decomposition* (SVD) of the weighted matrix. Thereby the original matrix A is decomposed as follows:

$$SVD(A) = U\Sigma V^T \quad (1)$$

The matrices U and V consist of the eigenvectors of the columns and rows of A . Σ is a diagonal matrix, containing in descending order the singular values of A . By only keeping the k strongest (k usually being 100 to 300) eigenvectors of either U or V , a so-called semantic space can be constructed for the terms or the contexts, respectively. Each term or each context then corresponds to a vector of k dimensions, whose distance to others can be compared by a standard vector distance measure. In most LSA approaches the *cosine* measure is used.

We use a slightly different setting, close to the one described by Schütze (1998) and Cederberg & Widdows (2003), where the original matrix is not based on occurrences of terms in documents, but on other co-occurring terms (term \times term-matrix). We thus count the frequency with which a given term occurs with others in a predefined context window ($\pm 10 - 100$ words). After applying *singular value decomposition*, each word is represented as a vector of k dimensions, and for every word pair w_i, w_j of our vocabulary we can calculate a similarity value $Sim(w_i, w_j)$, based on the *cosine* between their respective vectors.

3.2 Local resource confidence (LRC)

When combining relations from different sources, not all of them will be equally reliable. Depending on the type of resource in question, relations can be already equipped with a confidence value. For example, an extraction technique matching lexico-syntactic patterns

on the web counts the number of matches for two words w_i and w_j and a given pattern π ($[w_i \pi w_j]$). When this value is normalized by the maximum frequency of $[w_i \pi]$, each extracted relation triple t_k in resource r can be assigned a local resource confidence value $LRC(t_{kr})$ between 0 and 1. The following list defines the *local* confidence ratings that we use for the resources incorporated so far:

- *Wiktionary*: relative frequency of relation (frequency of a relation / number of relations)
- *OpenThesaurus*: relative frequency of relation
- *Wortschatz*: relative frequency of relation
- *Transl. WordNet*: mean translation confidence. Given a translation t_1 for a WN synset s_1 with a reliability $r_1 \in [0, 1]$ and a translation t_2 for synset s_2 with a reliability r_2 , and given $R(s_1, s_2)$ in WordNet we set the *local resource confidence* of $R(t_1, t_2)$ to the mean of r_1 and r_2 .
- *Hearst patterns*: maximum likelihood. Given two terms w_1 and w_2 , matched by a pattern π ($w_1\pi w_2$), we divide the matching frequency of $w_1\pi w_2$ with the frequency of $w_1\pi$

Even though ranging between 0 and 1, we acknowledge that the mathematical properties as well as the semantics of these measures are difficult to compare. However, we prefer to exploit the confidence ratings provided by the resources themselves than to assume uniform confidence for every incoming relation.

3.3 Global resource confidence (GRC)

A hand-coded resource like *Wiktionary* is surely more trustworthy than automated extraction techniques, which yield usually rather noisy results. A relation coming from *Wiktionary* should therefore receive a higher overall confidence than one coming from a pattern-based approach. Estimating the overall confidence of a resource can be done by determining the average LSA similarity for all n word pairs w_i, w_j figuring in the relation triples t_k of the resource r . A high *GRC* value (formula 2) indicates that the terms connected via relations in that resource fall into one semantic field in real life texts. Table 2 shows *GRC* values for the resources we have integrated so far in LEXO. A reference LSA space was calculated on a 101 million word corpus consisting of German *Wikipedia* and newspaper articles from a German daily (*Die Tageszeitung*, 1996 – 1999) and then reduced to 150 dimensions. For the calculation we used the *Infomap* toolkit, v0.8.6⁶, the co-occurrence window was set to ± 100 words.

$$GRC_r = \frac{1}{n} \sum_{k=0}^n Sim_{LSA}(w_i, w_j) \quad (2)$$

⁶<http://infomap-nlp.sourceforge.net/>

Source	raw	norm.	human	Dev.
<i>Wiktionary</i>	0.163	0.74	70%	+4%
<i>OpenThes.</i>	0.147	0.68	74%	-6%
<i>Hearst-p.</i>	0.109	0.51	57%	-6%
<i>transl. WN</i>	0.087	0.41	40%	+1%
<i>Wortschatz</i>	0.138	0.62	40%	+22%

Table 2: LSA based confidence values and human judgements for different resources.

To evaluate the accuracy of the calculated *GRC* values, we drew for each resource a random test sample of 100 triples. These triples were manually evaluated by 3 human annotators. We asked the annotators simply to label if the given relation holds or not (e.g. "Is X a hyponym for Y?"). The percentages of correct relations, as judged by the annotators, are also given in table 2.

As can be seen immediately, these results correlate strongly with the *GRC* values, with one exception: The *GRC* value of the *Wortschatz* data is obviously overestimated by our automatic measure. This is due to an apparent weakness of LSA, which is not able to distinguish between the relation types. Further manual inspection showed that the *Wortschatz* data contain mostly relations which would more appropriately be labeled as "near"-synonyms or loose associations, not as true synonyms. The goal of our project is to rely as little as possible on manual human inspection, but so far, our *GRC* measure has no means to detect relation mislabeling. For this reason we use meanwhile for the *Wortschatz* data a corrected *GRC* value (0.40), and we will try to develop more sophisticated measures in order to better estimate the reliability of a resource.

3.4 Confidence from structural information

For the estimation of confidence for a given relation we can not only exploit information inherent to the relation and its resource, but also on evidence from the already assembled data. One would probably assume that a *synonym* relation (x, y) is more reliable, if we have already the inverse relation (y, x) in the data base. Likewise, if we find for a given *hyponym* relation (x, y) its inverse *hypernym* pair (y, x) (this counts also for *mero-* and *holonyms*), we want to give it a higher confidence rating. Finally, due to the (normally assumed) transitivity of hyponymy, if we find for a *hyponym* pair (x, y) also the *hyponym* pairs (y, z) and (x, z) , we can assume (x, y) to be more reliable.

To make use of this kind of information, we define a range of indicator functions I_{1-4} returning 1, if one of the following conditions holds for a given triple $R(x, y)$, and 0 else.

- Synonym symmetry:**
 $I_1 = \text{syn}(x, y) \wedge \text{syn}(y, x)$
- Hypo-/hypernym correspondence:**
 $I_2 = \text{hypo}(x, y) \wedge \text{hyper}(y, x)$
- Mero-/holonym correspondence:**
 $I_3 = \text{mero}(x, y) \wedge \text{holo}(y, x)$
- Hypernym commonness:**
 $I_4 = \text{hypo}(x, y) \wedge \text{hypo}(x, z) \wedge \text{hypo}(y, z)$

The list of indicator functions is not meant to be exhaustive, there might be many more of such conditions playing a role in confidence estimation.

3.5 Individual semantic similarity

As long as we regard semantic relations, we can assume that the terms w_{k1} and w_{k2} of a triple t_k have

a high semantic similarity as calculated by a method like *LSA*. This gives us another confidence measure for a given triple t_k :

$$\text{Sim}(t_k) = \nu \cdot (\cos_{LSA}(w_{k1}, w_{k2})) \quad (3)$$

The factor ν normalizes the result, so that it also ranges between 0 and 1.

3.6 Integrated confidence

When we integrate the resources, we combine all single confidence values by linear interpolation. The *LRC* values ($LRC(t_{kr})$) of all resources for a relation are accumulated, according to the overall confidence GRC_r of the respective resource r . We then add the structural confidence and the semantic similarity score *Sim*.

$$\begin{aligned} IC(t_k) = & \lambda_1 \cdot \left(\nu \sum_{r=0}^n GRC_r \cdot LRC(t_{kr}) \right) \quad (4) \\ & + \lambda_2 \cdot I_1(t_k) \\ & + \lambda_i \cdot I_j(t_k) \dots \\ & + \lambda_m \cdot \text{Sim}(t_k) \end{aligned}$$

After integration, every relation triple t_k has an integrated confidence value *IC*, calculated from the single confidence values of the resources, where t_k appeared, weighted by their respective *GRC* value, the structural confidence functions $I_i(t_k)$ and the similarity function *Sim*(t_k). $\lambda_{1..n}$ are the coefficients controlling the importance of each component and sum up to 1. They can be optimized by an *EM*-style algorithm (cf. Dempster et al. 1977). ν is a normalizing factor, assuring that the accumulated confidence scores remain between 0 and 1 and n the number of resources integrated so far.

4 Syntactic integration

After a new set of relation hypotheses has been collected from external sources, these data have to be added to the already cumulated lexico-semantic resource (which is empty in the first iteration). In this step we have to solve two main problems in order to create an integrated and consistent data set: unification of word senses and resolution of possible inconsistencies.

4.1 Dealing with word senses

One of the major problems in combining lexical data from different resources lies in the discrimination of word senses (WS). If the only identifier of a term is its lexical form, it is impossible to automatically distinguish polysemous words. This is not only impractical for many applications, it also leads to weird constructions in the resulting ontology. Suppose a data set contains the following triples:

hyponym(*Tree*, *Plant*)
hyponym(*Tree*, *Structure*)
hyponym(*Oak*, *Tree*)

Due to the transitivity of the relation *hyponym* an automatic reasoner would infer here that an oak is both a plant and a structure. Obviously, the identifier *Tree* needs to be split (e.g. *Tree*₁ for the *plant* sense, and *Tree*₂ for *structure*).

Fortunately, some of the resources that we are using (e.g. *Wiktionary* and *OpenThesaurus*) do distinguish WS, but most other data sets (esp. from automatic extraction methods) do not support WS distinction.

This problem of data integration is close to the problem of mapping from a lexical resource to an ontology (or to another lexical resource). This issue is discussed in the literature (cf. Niles & Pease 2003), however, no general mapping strategy is available. In LEXO, we use corpora-based methods and contexts of terms in data sets for WS disambiguation. We define a *context* for a term t in a resource r as a set of all terms⁷ that co-occur with t in triples from r (or co-occur with terms that co-occur with t). A similar method was used for example by Buitelaar & Sacaleanu (2001). The transitivity of hyponymy and meronymy is used to extend a context of a term t with all "ancestors" of t . The word senses of terms in the context sets are ignored, because every context set is supposed to define proper WS of its members.

Given a set of triples S_1 where the word senses are distinguished, another set of triples S_2 has to be integrated with S_1 . Let us first consider the case when S_2 distinguishes between word senses. We illustrate this case by examples from *Wiktionary* and *OpenThesaurus*, presented in table 1. The term *Stern* ('star') is polysemous in both resources. The relations of this term are used to build its context. The context of *Stern*₁ in *OpenThesaurus* is *Asterisk*, *Asteriskus*, *Sternchen* and the context of *Stern*₂ in *Wiktionary* is *Asterisk*, *Sternchen*, *Symbol*. Since these contexts overlap (*Asterisk*, *Sternchen*), they are supposed to define the same WS. Thus, *Stern*₁ and *Stern*₂ are unified to *Stern*₁ in the resulting integrated data set. Resources may contain not enough information for word sense unifying (e.g. for *Stern*₃ and *Stern*₄ in our example). In this case it is necessary to refer to external information sources (cf. for example Dorow & Widows 2003), or a method like LSA (cf. 3) providing a similarity measure for the contexts of *Stern*₃ and *Stern*₄.

If a resource to be added does not distinguish between word senses, then every term from this set has to be considered as potentially polysemous. Let us consider the triples extracted from our *Wortschatz* data. We cannot use the information about the combined context of *Stern* anymore and have to treat every triple separately. For example, if a triple *synonym*(*Stern*, *Filmstar*) ('star', 'movie star') is to be added, the context of *Stern* in this case will be limited to *Filmstar*. Again, an LSA-based method can be used to measure the similarity between the term *Filmstar* and the contexts of *Stern* in the integrated data set (*Asterisk*, *Symbol*, ... ('asterisk', 'symbol'), *Himmelskörper*, *Gestirn*, ... ('heavenly body' sense) and *Mensch*, *Star* ('person' sense)).

4.2 Formulating Consistency Conditions

An important benefit of using a formalized ontological database in applications is the possibility to reason over the content of the ontology. For example, the inference of a subsumption hierarchy may help in formulating selectional restrictions, disambiguation tasks etc. But if the ontology contains mistakes and inconsistencies, reasoning may appear to be misleading and therefore pointless. There is a lot of literature on logical inconsistencies in ontological knowledge bases (cf. Kalyanpur 2006). However, as far as

⁷At present we consider only the most general lexical-semantic relations (synonymy, hyponymy, meronymy, antonymy). If more specific relations will be added, a new methodology of constructing term contexts can turn out to be necessary.

we know, no consistency constraints have been formulated yet for lexical resources (such as WordNet).

As we do not make use of complex logical statements (such as number restrictions, role inclusion, etc.) our resulting ontology is simple from a logical point of view.⁸ Still, it should obey certain structural criteria: For example, we do not want to allow that two or more semantic relations hold between a term pair (e.g. *synonym*(w_1, w_2) and *hyponym*(w_1, w_2)). Another structure that should be avoided are cycles; cyclic definitions may occur, when one resource claims that w_1 is a direct or indirect hyponym of w_2 while another resource contains w_2 as a hyponym of w_1 . After the unification of word indices is completed, the resulting hypothesis base is checked for consistency. Some examples of the constraints are given below (x, y stand for word senses, r stands for a relation).

1. Anti-reflexivity:

$$\forall x, y, r : r(x, y) \wedge r(y, x) \rightarrow x = y$$

2. Relation uniqueness:

$$\forall x, y, r_1, r_2 : r_1(x, y) \wedge r_2(x, y) \rightarrow r_1 = r_2$$

3. Transitivity:

$$\forall x, y, r : r \in Trans \wedge r(x, y) \wedge r(y, x) \rightarrow x = y$$

The *anti-reflexivity* constraint claims that terms are not allowed to be connected with themselves. Explicit reflexivity of synonymy is just redundant whereas reflexivity of some other relations (e.g. antonymy, hyponymy, meronymy) is wrong. The *relation uniqueness* constraint claims that only one relation can hold between two word senses. The *transitivity* constraint ensures that for relations that are declared to be transitive (i.e. antisymmetric) no cycles occur.

In our framework, inconsistency is resolved by ranking the axioms provoking the inconsistency by their confidence score. If a relation triple provokes more than one inconsistency then its ranking will be decreased. The relations with the lowest scores are then iteratively excluded until the inconsistency is resolved. If two candidates for exclusion have an equal ranking then the relation triple the removal of which entails less information loss (checked via inferences) will be eliminated.

Since the WS unification step in our project has not been finished yet, we cannot report about the overall inconsistencies in the integrated structure. However, a preliminary inconsistency evaluation of every single data source is available. For example, 1426 term pairs connected with more than one relation were found in *OpenThesaurus*; *Wiktionary* contains 1696 such pairs; in the *Wortschatz* data no such pairs have been found.

The list of the inconsistency constraints is still open. Probably some more constraints will be identified and added after the first evaluation of the resulting integrated resource has been completed.

5 The LexO architecture

The overall architecture of the LEXO framework is displayed in figure 1. We can distinguish three parts: On the lefthand side we find all incoming resources. They provide hypotheses in form of relation triples, which are then integrated by the system. The LEXO *engine* (middle) manages the hypothesis database (including confidence values and history for each entry) and its translation to an ontology. On the righthand side we find the output interfaces: A web access and

⁸Due to the lack of negation in the relations, the resulting structure cannot become *logically* inconsistent. We rather refer to *structural consistency* here.

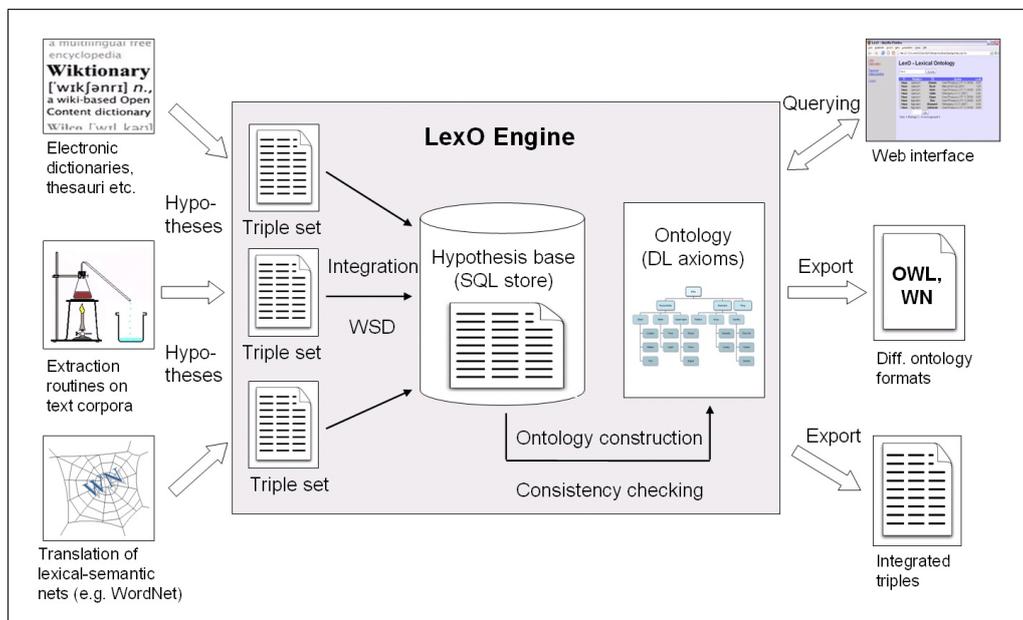


Figure 1: The LEXO Architecture

several export routines converting the data to different output formats.

5.1 The hypothesis database

In LEXO, a hypothesis is a lexical relation found in some of the various resources that can be used as input. An entry in this database contains the following data:

- The relation itself as a triple (**word-1**, **relation**, **word-2**). If the resource provides sense distinction (as e.g. *Wiktionary* or *OpenThesaurus*), the sense indices are kept together with **word-1** and **word-2**.
- A description of the source of the hypothesis
- A confidence value for the hypothesis (calculated as shown in 3)
- A timestamp indicating when the hypothesis was added

This organisation of the hypothesis base allows for an incremental adding of new hypotheses as well as revision and versioning. For any given point in time the state of the hypothesis base can be reconstructed so that the circumstances leading to a decision in the ontology construction process can be analysed.

5.2 The LexO engine

The LEXO engine is the central part of the framework. It manages the database, provides facilities for integrating, filtering and cleaning the raw data (relation triples) and builds up a structured representation assuring pre-defined consistency criteria.

The main problem in the translation process is the confidence-based selection of relations. All data is considered to be more or less reliable (cf. section 3), but, apart from the sanity conditions described in 4.2 (relation uniqueness, connectedness, acyclicity etc.) we have no absolute reliability criterion. We therefore apply a heuristic threshold on the confidence values, depending on the overall growth of the ontology.

5.3 Import-/export interfaces

LEXO provides a library of import and export functions as well as a set of interfaces based on it. A number of scripts have been developed to convert each of the resources to triple sets (with *a-priori* confidence values, depending on the resource), and possibly word sense distinction (if provided by the resource). After conversion, a script deals then with the integration of the triples, word sense unification and the import to the triple store (SQL database). In this step, the confidence values are updated, according to the method described in section 3.

The database as well as the ontology can be queried via a web interface (online soon!). This interface will provide masks that allow to search for individual words and relations. Furthermore, another set of converting tools will allow to export the ontology into different formats such as a set of *OWL* clauses or as a *WordNet*-like database. Methods how to achieve a reasonable *OWL* representation of lexical-semantic relations have been presented by van Assem et al. (2004) and Huang & Zhou (2007). Since plain relations can also be of interest for many applications, a database dump of the hypothesis base will also be provided.

5.4 Evaluation scenarios

Since our project is still in its beginning, we cannot offer any real evaluation of the data yet. However we want to describe here, how an evaluation can be performed. There are basically three complementary strategies: The first is widely used in this domain, because it is straightforward and quick; it is used, for example, by Cimiano et al. (2005). The presupposition is here that we have a reference ontology at hand (gold standard), to which we can compare our data. In the simplest form, we then measure the overlap of relations between our data set and the reference resource in terms of recall and precision. There exist also more complex measures taking the structural similarity into account (cf. Dellschaft & Staab 2006). Our reference resource could be, for example *GermaNet*, the German word net. However, by determining the overlap of our data with *GermaNet*, we evaluate obviously not the overall quality, but foremost the

similarity with GermaNet, which is a questionable aspect.

The second strategy relies on direct inspection of the data, it was used, for example, by Snow et al. (2006). Here, human annotators evaluate a representative sample of the constructed data set. Whereas this approach can be very accurate, given the sample is sufficiently large, it implies a lot of efforts and cannot be used for the optimization of confidence parameters (cf. section 3), for example.

The third evaluation scenario is an indirect one. Since the main aim of our project is to serve as a structured semantic resource for NLP tasks, we can evaluate its quality by assessing its performance herein. Harabagiu & Moldovan (2000) for example assess their enriched taxonomy on three tasks: word-sense disambiguation, coreference resolution and information extraction. Measuring the performance of an ontology in such a way implies of course a lot of effort, but it is an objective and independent measure. For this reason we favor this strategy for evaluating the quality of our data.

6 Conclusion and future work

We have proposed an architecture for collecting and integrating lexical-semantic data from various resources. All incoming relations are stored as hypotheses in a database, annotated with automatically determined confidence values. An ontology is created from this hypothesis base by interpreting certain lexical-semantic relations as ontological statements.

We claim that this approach proves especially useful when a broad range of different resources is combined. Therefore we plan to implement additional extraction methods to open up new sources of lexical-semantic information. Beside new sources we will also integrate more types of relations into the database. Apart from that, future efforts will tackle the following issues:

Parameter and threshold estimation: Our project is still in the stage of data cumulation. Whereas we have described in 3, how confidence values can be determined for each relation, we have not optimized the necessary parameters yet. Moreover, we have not yet determined a reasonable threshold for the confidence scores. This kind of parameter tuning takes a lot of time and work and will be subject to our coming efforts.

Creating a common data structure using a top-level ontology: In order to create an ontologically uniform data structure, we want to use a hand-crafted top-level ontology as a seeding ground onto which relations from the database will be successively added. By predefining the top-level concepts we have a means to influence the overall growth of the resulting ontology. However, since the further evolution of the structure strongly depends on the ontological properties of the top-level categorization, it is crucial to construct this structure with a lot of care. Here, the work of Guarino (1998), Gangemi et al. (2002) and Guarino & Welty (2004) will provide valuable guidelines.

Implementation of converters and interfaces: In section 5 we described the output interfaces of our system. These are not implemented yet, but we will try to provide a usable web interface including the possibility to download our data shortly.

Structural constraints and axiomatization: In section 4.2 several simple consistency conditions for our lexical ontology were formulated. However, this set of constraints is definitely not exhaustive. In order to define which constraints are necessary and sufficient for achieving the proposed goals, we need to develop a precise axiomatization of relations and top-level categories in LexO (cf. Gangemi et al. 2001).

7 Acknowledgements

This work is supported by the *Deutsche Forschungsgemeinschaft*, research group “Text Technology” (FOR 437, project C2). The authors also want to thank Helmar Gust, Universität Osnabrück, for fruitful discussions on the subject.

References

- Agichtein, E. & Gravano, L. (2000), Snowball: Extracting relations from large plain-text collections, in ‘Proc. of the 5th ACM International Conference on Digital Libraries (ACM DL)’, pp. 85–94.
- Atserias, J., Climent, S., Farreres, X., Rigau, G. & Rodriguez, H. (1997), Combining Multiple Methods for the Automatic Construction of Multilingual WordNets, Technical report, Departament de Llenguatges i Sistemes Informatics, Universitat Politècnica de Catalunya, Barcelona.
- Brin, S. (1998), Extracting patterns and relations from the world wide web, in ‘WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT’98’.
- Buitelaar, P., Olejnik, D., Hutanu, M., Schutz, A., Declerck, T. & Sintek, M. (2004), Towards ontology engineering based on linguistic analysis, in ‘Proc. of the Lexical Resources and Evaluation Conference’.
- Buitelaar, P., Olejnik, D. & Sintek, M. (2004), A Protégé plugin for ontology extraction from text based on linguistic analysis, in ‘Proc. of the 1st European Semantic Web Symposium (ESWS)’.
- Buitelaar, P. & Sacaleanu, B. (2001), Ranking and selecting synsets by domain relevance, in ‘Proc. of NAACL’01’.
- Caraballo, S. (1999), Automatic construction of a hypernym-labeled noun hierarchy from text, in ‘Proc. of the 37th Annual Meeting of the Association for Computational Linguistics’, pp. 120–126.
- Cederberg, S. & Widdows, D. (2003), Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy, in ‘Proc. of the Conference on Natural Language Learning’.
- Cimiano, P. (2006), *Ontology learning and population from text. Algorithms, Evaluation and Applications*, Springer.
- Cimiano, P., A., P., Schmidt-Thieme, L. & Staab, S. (2005), *Learning Taxonomic Relations from Heterogenous Sources of Evidence*, IOS Press, pp. 59–73.
- Cimiano, P. & Staab, S. (2004), ‘Learning by googling’, *SIGKDD Explorations* 6(2).
- Cimiano, P. & Staab, S. (2005), Learning concept hierarchies from text with a guided agglomerative clustering algorithm, in ‘Proc. of the ICML Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods’, Bonn, Germany.

- Cimiano, P. & Völker, J. (2005), Text2Onto - a framework for ontology learning and data-driven change discovery, in 'Proc. of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB'2005)'.
- Cruse, D. A. (1986), *Lexical Semantics*, Cambridge University Press.
- Deerwester, S., Dumais, S., Furnas, G., Landauer, T. & Harshman, R. (1990), 'Indexing by Latent Semantic Analysis', *JASIS* 41(6), pp. 391–407.
- Dellschaft, K. & Staab, S. (2006), On how to perform a gold standard based evaluation of ontology learning, in I. C. et al., ed., 'Proc. of the 5th International Semantic Web Conference (ISWC)', LNCS 4273, Springer Verlag, pp. 228–241.
- Dempster, A., Laird, N. & Rubin, D. (1977), 'Maximum likelihood from incomplete data via the em algorithm', *Journal of the Royal Statistical Society* 39(1), 1–38.
- Dorow, B. & Widdows, D. (2003), Discovering corpus-specific word senses, in 'Proc. of EACL', Budapest, Hungary., pp. 79–82.
- Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popesu, A.-M., Shaked, T., Soderland, S., S., W. D. & Yates, A. (2004), Web-scale information extraction in KnowItAll, in 'Proc. of the 13th World Wide Web Conference', pp. 100–110.
- Faure, D. & Nédellec, C. (1998), ASIUM: Learning subcategorization frames and restrictions of selection., in 'Proc. of the 10th Conference on Machine Learning (ECML)'.
- Fellbaum, C., ed. (1998), *WordNet: An Electronic Lexical Database*, MIT Press.
- Gangemi, A., Guarino, N., Masolo, C. & Oltramari, A. (2001), Understanding top-level ontological distinctions, in 'Proc. of IJCAI'01'.
- Gangemi, A., Guarino, N., Masolo, C., Oltramari, A. & Schneider, L. (2002), Sweetening ontologies with DOLCE, in 'Proc. of EKAW'02'.
- Guarino, N. (1998), Some ontological principles for designing upper level lexical resources, in 'Proc. of the First International Conference on Lexical Resources and Evaluation', Granada, Spain.
- Guarino, N. & Welty, C. (2004), *The Handbook on Ontologies*, Springer-Verlag, chapter An overview of OntoClean, pp. 151–172.
- Hamp, B. & Feldweg, H. (1997), Germanet - a lexical-semantic net for german, in 'Proc. of the ACL workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications', Madrid.
- Harabagiu, S. & Moldovan, D. (2000), *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, AAAI/MIT Press, chapter Enriching the WordNet Taxonomy with Contextual Knowledge Acquired from Text, pp. 301–334.
- Hearst, M. A. (1992), Automatic acquisition of hyponyms from large text corpora, in 'Proc. of the 14th Int. Conf. on Computational Linguistics', Nantes, France.
- Huang, X. X. & Zhou, C. L. (2007), 'An OWL-based WordNet lexical ontology', *Journal of Zhejiang University* 8(6), 864–870.
- Kalyanpur, A. (2006), Debugging and Repair of OWL Ontologies, PhD thesis, University of Maryland College Park.
- Katrenko, S. & Adriaans, P. (2006), Learning patterns from dependency paths, in 'Proc. of the international workshop ontologies in text technology (OTT'06)', Osnabrück.
- Khan, L. R. & Hovy, E. H. (1997), Improving the Precision of Lexicon-to-Ontology Alignment Algorithms, in 'Proc. of the 1st AMTA Workshop on Interlinguas'.
- Knight, K. & Luk, S. K. (1994), Building a Large-Scale Knowledge Base for Machine Translation, in 'Proc. of the American Association of Artificial Intelligence AAAI-94.', Seattle, WA., pp. 773–778.
- Landauer, T. K. & Dumais, S. T. (1997), 'A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge', *Psychological Review* 104(1), 211–240.
- Maedche, A. (2002), *Ontology learning for the semantic web*, Kluwer.
- Naber, D. (2005), OpenThesaurus: ein offenes deutsches Wortnetz, in 'Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen: Beiträge zur GLDV-Tagung 2005.', Peter Lang Verlag, Bonn, pp. 422–433.
- Navigli, R. & Velardi, P. (2004), 'Learning domain ontologies from document warehouses and dedicated websites', *Computational Linguistics* 30(2).
- Niles, I. & Pease, A. (2003), Linking lexicons and ontologies: Mapping WordNet to the suggested upper merged ontology, in 'Proc. of IKE'03, Las Vegas'.
- Okumura, A. & Hovy, E. (1994), Lexicon-to-Ontology Concept Association Using a Bilingual Dictionary, in 'Proc. of AMTA, Columbia, MD, 6-8 oct.'.
- Oltramari, A., Gangemi, A., Guarino, N. & Masolo, C. (2002), Restructuring WordNet's top-level: The OntoClean approach, in 'Proc. of LREC2002 (OntoLex Workshop)'.
- Salton, G. & McGill, M. (1983), *Introduction to Modern Information Retrieval*, McGraw-Hill, New York.
- Schütze, H. (1998), 'Automatic word sense discrimination', *Computational Linguistics* 24(1), pp. 97–124.
- Snow, R., Jurafsky, D. & Ng, A. Y. (2006), Semantic taxonomy induction from heterogenous evidence, in 'Proc. of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL', Association for Computational Linguistics, Morristown, NJ, USA, pp. 801–808.
- van Assem, M., Menken, M., Schreiber, G., Wielmaker, J. & Wielinga, B. (2004), A method for converting thesauri to rdf/owl, in 'Proc. of the 3rd Int. Semantic Web Conference (ISWC)', Hiroshima, Japan.
- Wagner, A. (2000), Enriching a lexical semantic net with selectional preferences by means of statistical corpus analysis, in 'Proc. of the ECAI Workshop on Ontology Learning', Berlin, Germany.
- Wandmacher, T. (2005), How semantic is Latent Semantic Analysis?, in 'Proc. of TALN/RECITAL'05', Dourdan, France.

Author Index

Abdul Kareem, Sameem, 35
Akand, Elma, 15
Amaral, Fernando N., 55

Bahreini, Kiavash, 7
Bain, Michael, 15
Bazilio, Carlos, 55
Boardman, Glenn, 25

Corbett, Dan, 45
Costa, Vaston, 55

da Silva, Geiza M.H., 55
de Vasconcelos, Davi Romero, 55
Dittmann, Henrik, 61

Elçi, Atilla, 7

Haeusler, Edward Hermann, 55

Hajmoosaei, Abdolreza, 35
Hunter, Jane, 3

Krumnack, Ulf, 61

Lu, Hongen, 25

Meyer, Thomas, iii

Nayak, Abhaya C., iii
Nguyen, Philip H.P., 45

Ovchinnikova, Ekaterina, 61

Rademaker, Alexandre, 55

Temple, Mark, 15

Wandmacher, Tonio, 61

Recent Volumes in the CRPIT Series

ISSN 1445-1336

Listed below are some of the latest volumes published in the ACS Series *Conferences in Research and Practice in Information Technology*. The full text of most papers (in either PDF or Postscript format) is available at the series website <http://crpit.com>.

- Volume 67 - Conceptual Modelling 2007**
 Edited by John F. Roddick, *Flinders University* and Annika Hinze, *University of Waikato, New Zealand*. January, 2007. 978-1-920682-48-4.
 Contains the proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling (APCCM2007), Ballarat, Victoria, Australia, January 2007.
- Volume 68 - ACSW Frontiers 2007**
 Edited by Ljiljana Brankovic, *University of Newcastle*, Paul Coddington, *University of Adelaide*, John F. Roddick, *Flinders University*, Chris Steketee, *University of South Australia*, Jim Warren, *the University of Auckland*, and Andrew Wendelborn, *University of Adelaide*. January, 2007. 978-1-920682-49-1.
 Contains the proceedings of the ACSW Workshops - The Australasian Information Security Workshop: Privacy Enhancing Systems (AISW), the Australasian Symposium on Grid Computing and Research (AUSGRID), and the Australasian Workshop on Health Knowledge Management and Discovery (HKMD), Ballarat, Victoria, Australia, January 2007.
- Volume 69 - Safety Critical Systems and Software 2006**
 Edited by Tony Cant, *Defence Science and Technology Organisation, Australia*. February, 2007. 978-1-920682-50-7.
 Contains the proceedings of the 11th Australian Conference on Safety Critical Systems and Software, August 2006, Melbourne, Australia.
- Volume 70 - Data Mining and Analytics 2007**
 Edited by Peter Christen, Paul Kennedy, Jiuyong Li, Inna Kolyshkina and Graham Williams. December, 2007. 978-1-920682-51-4.
 Contains the proceedings of the 6th Australasian Data Mining Conference (AusDM 2007), Gold Coast, Australia, December 2007.
- Volume 72 - Advances in Ontologies 2006**
 Edited by Mehmet Orgun, *Macquarie University* and Thomas Meyer, *National ICT Australia, Sydney*. December, 2006. 978-1-920682-53-8.
 Contains the proceedings of the Australasian Ontology Workshop (AOW 2006), Hobart, Australia, December 2006.
- Volume 73 - Intelligent Systems for Bioinformatics 2006**
 Edited by Mikael Boden and Timothy Bailey, *University of Queensland*. December, 2006. 978-1-920682-54-5.
 Contains the proceedings of the AI 2006 Workshop on Intelligent Systems for Bioinformatics (WISB-2006), Hobart, Australia, December 2006.
- Volume 74 - Computer Science 2008**
 Edited by Gillian Dobbie, *University of Auckland, New Zealand* and Bernard Mans, *Macquarie University*. January, 2008. 978-1-920682-55-2.
 Contains the proceedings of the Thirty-First Australasian Computer Science Conference (ACSC2008), Wollongong, NSW, Australia, January 2008.
- Volume 75 - Database Technologies 2008**
 Edited by Alan Fekete, *University of Sydney* and Xuemin Lin, *University of New South Wales*. January, 2008. 978-1-920682-56-9.
 Contains the proceedings of the Nineteenth Australasian Database Conference (ADC2008), Wollongong, NSW, Australia, January 2008.
- Volume 76 - User Interfaces 2008**
 Edited by Beryl Plimmer and Gerald Weber, *University of Auckland*. January, 2008. 978-1-920682-57-6.
 Contains the proceedings of the Ninth Australasian User Interface Conference (AUI2008), Wollongong, NSW, Australia, January 2008.
- Volume 77 - Theory of Computing 2008**
 Edited by James Harland, *RMIT University* and Prabhu Manyem, *University of Ballarat*. January, 2008. 978-1-920682-58-3.
 Contains the proceedings of the Fourteenth Computing: The Australasian Theory Symposium (CATS2008), Wollongong, NSW, Australia, January 2008.
- Volume 78 - Computing Education 2008**
 Edited by Simon, *University of Newcastle* and Margaret Hamilton, *RMIT University*. January, 2008. 978-1-920682-59-0.
 Contains the proceedings of the Tenth Australasian Computing Education Conference (ACE2008), Wollongong, NSW, Australia, January 2008.
- Volume 79 - Conceptual Modelling 2008**
 Edited by Annika Hinze, *University of Waikato, New Zealand* and Markus Kirchberg, *Massey University, New Zealand*. January, 2008. 978-1-920682-60-6.
 Contains the proceedings of the Fifth Asia-Pacific Conference on Conceptual Modelling (APCCM2008), Wollongong, NSW, Australia, January 2008.
- Volume 80 - Health Data and Knowledge Management 2008**
 Edited by James R. Warren, Ping Yu, John Yearwood and Jon D. Patrick. January, 2008. 978-1-920682-61-3.
 Contains the proceedings of the Australasian Workshop on Health Data and Knowledge Management (HDKM 2008), Wollongong, NSW, Australia, January 2008.
- Volume 81 - Information Security 2008**
 Edited by Ljiljana Brankovic, *University of Newcastle* and Mirka Miller, *University of Ballarat*. January, 2008. 978-1-920682-62-0.
 Contains the proceedings of the Australasian Information Security Conference (AISC 2008), Wollongong, NSW, Australia, January 2008.
- Volume 82 - Grid Computing and e-Research**
 Edited by Wayne Kelly and Paul Roe, *QUT*. January, 2008. 978-1-920682-63-7.
 Contains the proceedings of the Australasian Workshop on Grid Computing and e-Research (AusGrid 2008), Wollongong, NSW, Australia, January 2008.
- Volume 83 - Challenges in Conceptual Modelling**
 Edited by John Grundy, *University of Auckland, New Zealand*, Sven Hartmann, *Massey University, New Zealand*, Alberto H.F. Laender, *UFMG, Brazil*, Leszek Maciaszek, *Macquarie University, Australia* and John F. Roddick, *Flinders University, Australia*. December, 2007. 978-1-920682-64-4.
 Contains the tutorials, posters, panels and industrial contributions to the 26th International Conference on Conceptual Modeling - ER 2007.
- Volume 84 - Artificial Intelligence and Data Mining 2007**
 Edited by Kok-Leong Ong, *Deakin University, Australia*, Wenyuan Li, *University of Texas at Dallas, USA* and Junbin Gao, *Charles Sturt University, Australia*. December, 2007. 978-1-920682-65-1.
 Contains the proceedings of the 2nd International Workshop on Integrating AI and Data Mining (AIDM 2007), Gold Coast, Australia, December 2007.
- Volume 86 - Safety Critical Systems and Software 2007**
 Edited by Tony Cant, *Defence Science and Technology Organisation, Australia*. December, 2007. 978-1-920682-67-5.
 Contains the proceedings of the 12th Australian Conference on Safety Critical Systems and Software, August 2006, Adelaide, Australia.