Workshop on Human Language Technology for the Semantic Web and Web Services

Proceedings of the ISWC 2003 Workshop

Programme chairs:
Hamish Cunningham, Ying Ding, Atanas Kiryakov

October 20th 2003, Sanibel Island, Florida

Programme Committee

Alexander Maedche, Robert Bosch Gmbh, Germany
Aldo Gangemi, Laboratory for Applied Ontology, ISTC-CNR, Italy
Asun Gomez-Perez, Universidad Politecnica de Madrid, Spain
Christopher A. Welty, IBM Watson Research Center, USA
David Harper, Robert Gordon University, Aberdeen, UK
Diana Maynard, University of Sheffield, UK
Dieter Fensel, University of Innsbruck, Austria
Dieter Merkl, TU Vienna, Austria
Fabio Crestani, University of Strathclyde, UK
Jan Paralic, Technical University Kosice, Slovakia
John Davies, British Telecom, UK
Valentin Tablan, University of Sheffield, UK
John Tait, University of Sunderland, UK
Jon Patrick, Univeristy of Sydney, Australia
Kalina Bontcheva, University of Sheffield, UK
Maria Vargas-Vera, Open University, UK
Marin Dimitrov, OntoText Lab, Bulgaria
Paul Buitelaar, DFKI, GE
Robert Engels, CognIT, Norway
Steffen Staab, University of Karlsruhe, Germany
Vojtech Svatek, University of Economics, Prague, Czech Republic
Wim Peters, University of Sheffield, UK
York Sure, University of Karlsruhe, Germany
Yorick Wilks, University of Sheffield, UK

Endorsed by...

Contents

Towards Semantic Web Information Extraction

Borislav Popov, Atanas Kiryakov, Dimitar Manov, Angel Kirilov, Damyan Ognyanoff, Miroslav Goranov

Ontotext Lab, Sirma AI EOOD, 135 Tsarigradsko Shose, Sofia 1784, Bulgaria
{naso, borislav, damyan, mitac, angel, miro}@sirma.bg

Abstract. The approach towards Semantic Web Information Extraction (IE) presented here is implemented in KIM – a platform for semantic indexing, annotation, and retrieval. It combines IE based on the mature text engineering platform (GATE[1]) with Semantic Web-compliant knowledge representation and management. The cornerstone is automatic generation of named-entity (NE) annotations with class and instance references to a semantic repository.

Simplistic upper-level ontology, providing detailed coverage of the most popular entity types (Person, Organization, Location, etc.; more than 250 classes) is designed and used. A knowledge base (KB) with de-facto exhaustive coverage of real-world entities of general importance is maintained, used, and constantly enriched. Extensions of the ontology and KB take care of handling all the lexical resources used for IE, most notable, instead of gazetteer lists, aliases of specific entities are kept together with them in the KB.

A Semantic Gazetteer uses the KB to generate lookup annotations. Ontology-aware pattern-matching grammars allow precise class information to be handled via rules at the optimal level of generality. The grammars are used to recognize NE, with class and instance information referring to the KIM ontology and KB. Recognition of identity relations between the entities is used to unify their references to the KB. Based on the recognized NE, template relation construction is performed via grammar rules. As a result of the latter, the KB is being enriched with the recognized relations between entities. At the final phase of the IE process, previously unknown aliases and entities are being added to the KB with their specific types.

1 Introduction

The acquisition of masses of metadata for the web content would allow various Semantic Web applications to emerge and gain wide acceptance. Such applications would provide and use new access methods based on the associated metadata. The manual semantic authoring, although accurate and sometimes unavoidable, simply does not match the scale as well as the authoring and usage practices typical for the web content. The approach for automatic extraction of metadata is promising scalable, cheap, author-independent and (optionally) user-specific enrichment of the web

[1] General Architecture for Text Engineering (GATE), http://gate.ac.uk, leading NLP and IE platform developed at the University of Sheffield.

content. However, at present there is no technology available to provide automatic semantic annotation in conceptually clear, intuitive, scalable, and accurate enough fashion. Even more, there is no clear vision regarding the approach and model for generation and representation of such annotations.

This paper presents first a model for semantic content enrichment, which we name semantic annotation (section 2.) This model is implemented in a system called KIM and presented in the third section. Most attention is paid to the information extraction (IE[2]) approach used in KIM for automatic semantic annotation; discussed in section 4 with its processing components, KB resources, and resulting linking of NE references to the ontology and KB. Next, evaluation of the performance is presented in the fifth section followed by short overview of related work in section 6. The last section provides a conclusion and discussion on future work.

2 Semantic Annotation

The semantic annotation offered here is a specific metadata generation and usage schema targeted to enable new information access methods and extend existing ones. It is based on the hypothesis that the named entities[3] mentioned in the documents constitute important part of their semantics. Semantic annotation is also the task for/process of generating such metadata.

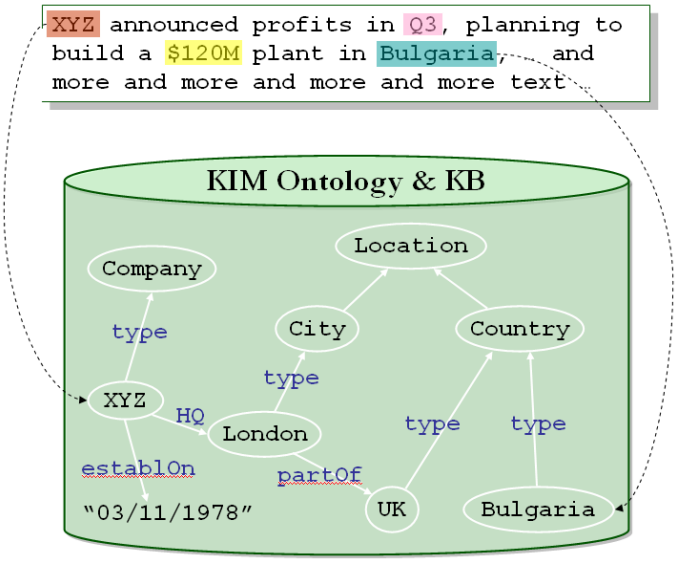


Fig. 1. Semantic Annotation

[2] Information Extraction is a relatively new discipline in the Natural Language Processing (NLP), which conducts partial analysis of text in order to extract specific information.

[3] Named Entities (NE) are people, organizations, locations, and others that are referred by name. The wide interpretation of the term includes any tokens referring something specific in the world: numbers, addresses, amounts of money, dates, etc.

In a nutshell, we consider Semantic Annotation the idea of assigning to the entities in the text links to their semantic descriptions (as presented on Fig. 1). The idea of this sort of metadata is to provide both class and instance information about the entities referred in the documents. It is a question of terminology whether these annotations should be called "semantic," "entity" or some other way. To the best of our knowledge there is no well established term for this task; neither there is a well established meaning for the term "semantic annotation"[4]. What is more important, the automatic semantic annotations enable many new applications: highlighting, indexing and retrieval, categorization, generation of more advanced metadata, smooth traversal between unstructured text and available relevant knowledge. Semantic annotation is applicable for any sort of text – web pages, regular (non-web) documents, text fields in databases, etc. Further, knowledge acquisition can be performed based on extraction of more complex dependencies – analysis of relationships between entities, event and situation descriptions, etc. We believe that, defined this way, semantic annotation is clearly specified, easy to understand, and can serve as a basis for number of useful applications (some of those demonstrated in KIM).

The automatic semantic annotation can be seen as a classical named-entity recognition (NER) and annotation process. The traditional flat NE type sets consist of several general types (such as **`Organization`**, **`Person`**, **`Date`**, **`Location`**, **`Percent`**, **`Money`**). Although these represent the most important domain-independent NE types, still the entities with same type are dividable in more specific classes from the average educated human (e.g. public companies, sport teams, and syndicates are all organizations). The semantic annotation is specific for providing more precise type information, because the NE type is specified by reference to an ontology. Further, and more important, the semantic annotation requires identification of the entity. While in a classical NER task, guessing the type is everything to be achieved, a semantic annotation needs to recognize the entity (either out of a set of known ones either as unknown one) and refer to it. There is some similarity with the understanding of "content extraction" as used in the context of the ACE project[5].

2.1 Semantic Annotation Model and Representation

Here we discuss the structure and the representation of the semantic annotations, including the necessary knowledge and metadata. There are number of basic prerequisite for representation of semantic annotations:

- Ontology (or at least taxonomy) bearing the classes of entities. It should be possible to refer to the classes in the ontology;
- Unique entity identifiers which allow, those to be identified and linked to their semantic descriptions;
- Knowledge base with entity descriptions.

[4] The term is previously used in [23] in a bit more general sense compared to what we propose, but it didn't get wide acceptance.

[5] See www.itl.nist.gov/iad/894.01/tests/ace/

The next question considers an important choice for the representation of the annotations – "to embed or not to embed?" There are number of arguments providing evidence that the semantic annotations have to be decoupled from the content they refer to. One key reason is to allow dynamic user-specific semantic annotations – the embedded annotations become part of the content and may not change corresponding to the interest of the user or the context of usage. Further, embedded complex annotations (such as those necessary for the Semantic Web) would have negative impact on the volume of the content and can complicate its maintenance – imagine that page with three layers of overlapping semantic annotations need to be updated preserving them consistent. Those and number of other issues defending the externally encoded annotation can be found in [20] which also provides an interesting parallel to the open hypermedia systems.

Once decided that the semantic annotations have to be kept separate from the content, the next question is whether or not (and how much) to couple the annotations with the ontology and the knowledge base? It is the case that such integration seems profitable – it would be easier to keep in synch the annotations with the class and entity descriptions. However, there are at least two important problems:

- Both the cardinality and the complexity of the annotations differ from those of the entity descriptions – the annotations are simpler, but their count is much bigger than this of the entity descriptions. Even considering middle-sized document corpora the annotations can reach tens of millions. Suppose 10M annotations are stored in an RDF(S) store together with 1M entity descriptions. Suppose also that each annotation and each entity description are represented with 10 statements. There is a big difference regarding the inference approaches and hardware capable in efficient reasoning and access to 10M-statement repository and with 110M-statement repository.
- It would be nice if the world knowledge (ontology and instance data) and the document-related metadata are kept independent. This would mean that for one and the same document different extraction, processing, or authoring methods will be able to deliver alternative metadata referring to one and the same knowledge store.
- Most important, it should be possible the ownership and the responsibility for the metadata and the knowledge to be distributed. This way, different parties can develop and maintain separately the content, the metadata, and the knowledge.

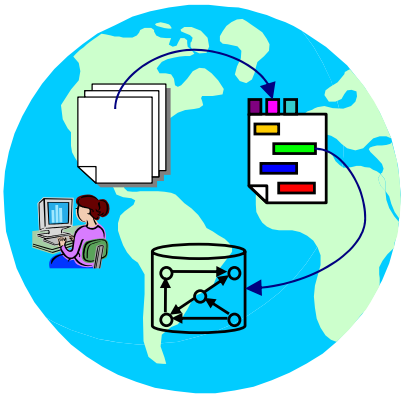


Fig. 2. Distributed Heterogeneous Knowledge

Based on the above arguments we propose decoupled representation and management of the documents, the metadata (annotations) and the formal knowledge (ontologies and instance data) as depicted on Fig. 2.

We will extremely shortly advocate the appropriateness of using ontology for defining the entity types – those are the only wide accepted paradigm for management of open, sharable, and reusable knowledge. According our view, light-weight ontology (poor on axioms) is sufficient for simple definition of the entity classes, their appropriate attributes, and relations. In the same time it allows more efficient and scalable management of the knowledge (compared to the heavy-weight semantic approaches.)

According to the analysis of ontology and knowledge representation languages and formats in [12] and other authors it becomes evident that there is no much consensus beyond RDF(S), see [4]. The latter is well established in the Semantic Web community as a knowledge representation and interchange language. The rich diversity of RDF(S) repositories, APIs and tools, forms a mature environment for development of systems grounded in RDF(S) representation of their ontological and knowledge resources. Because of the common acceptance of RDF(S) in the Semantic Web community, it would be easy to reuse the ontology and KB, as well as enrich them with domain-specific extensions. The new OWL (see [10]) standard offers clear, relatively consensual and backward-compatible path beyond RDF(S), but still lacks tool support. Our experience shows (see the section on KIM) that for the basic purposes of light-weight ontology definition and entity description, RDF(S) provides sufficient basic expressiveness. The most critical nice-to-have primitives (equality, transitive and symmetric relations, etc.) are well covered in OWL Lite – the simplest first level of OWL. So, we suggest that RDF(S) is used in a way which allows easy extension towards OWL[6] – this means avoiding primitives not included in the OWL schema.

3 The KIM platform

The KIM platform[7] provides semantic annotation, indexing, and retrieval services and infrastructure. The most important differences between KIM and other systems and approaches are that it performs semantic annotation and provides services based on the results. To do this in a consistent fashion, it performs information extraction based on an ontology and a massive knowledge base.

The traditional flat NE type sets consist of several general types (such as Organization, Person, Date, Location, Percent, Money). Although these represent the most important domain-independent NE types, still the entities with same type are dividable in more specific classes from the average educated human (e.g. public companies, sport teams, and syndicates are all organizations). We identified an inter-domain NE type hierarchy from a corpus of general news and integrated it in the KIM Ontology (KIMO). The ontology contains definitions of entities, relations, as well as a branch of lexical resource types (e.g. Title, PersonFirstName, DayOfWeek, etc.). The semantic descriptions of entities and relations between them are kept in a knowledge base (KB) encoded in the KIM ontology and residing in the same semantic repository. Thus KIM provides for each entity reference in the text (i) a link (URI) to the most

[6] http://www.w3.org/2002/07/owl

[7] Knowledge and Information Management Platform, see http://www.ontotext.com/kim

specific class in the ontology and (ii) a link to the specific instance in the KB. Each extracted NE is linked to its specific type information (thus Arabian Sea would be identified as **`Sea`**, instead of the traditional – **`Location`**). Also each NE is linked to an individual in the KB and the associated semantic description (attributes and relations of the entity). The KB has been pre-populated with entities of general importance, and is iteratively enriched with entity individuals and relations as a result of the IE process. Thus the extracted named entities could be further used for semantic indexing and retrieval of content with respect to entity instance and type. Thus allowing the satisfaction of requests that inquire for documents which refer entities described with type, name, and attribute restrictions, as well as the expected relations between these entities (e.g. look for a **`Sea`** that is a **`subRegionOf`** the Indian Ocean).

The information extraction process in KIM is based on the GATE platform. Few generic NLP components for tokenization, part-of-speech tagging, and others, have been directly reused by KIM. GATE's pattern-matching grammars have been modified to handle entity class information and allow generalization of the rules (e.g. specifying a pattern consisting of all **`Location`**s that are **`subRegionOf`** a **`Country`**, instead of specifying the concrete types of all the possible location sub-classes – **`City`**, **`Province`**, **`CapitalCity`**, etc.) The KIM gazetteer lookup component looks up entities and lexical resources by their aliases. The aliases present entity names or keys (suffixes, context words) and serve as clues for the pattern-matching grammar NER process. As a part of the KIM platform, the KIM IE is open towards the semantic repository that keeps the ontology and the KB, and depends on these for initialization of its processing components. Finally the IE identifies the instance information for each known NE in the text, and adds the new entities with their semantic descriptions and relations to the KB. Thus as a result each NE reference is linked to its type and its individual semantic description.

For the end-user, the KIM IE functionality is straightforward and simple – requesting annotation from a browser plug-in, which highlights the entities in the current content and generates a hyperlink used for further exploration of the available knowledge for the entity (as shown on Fig. 3). Various access methods are also available – entity pattern search, entity lookup, keyword and document attribute search. There is also an opportunity to create a composite query consisting of atomic searches of the above types.

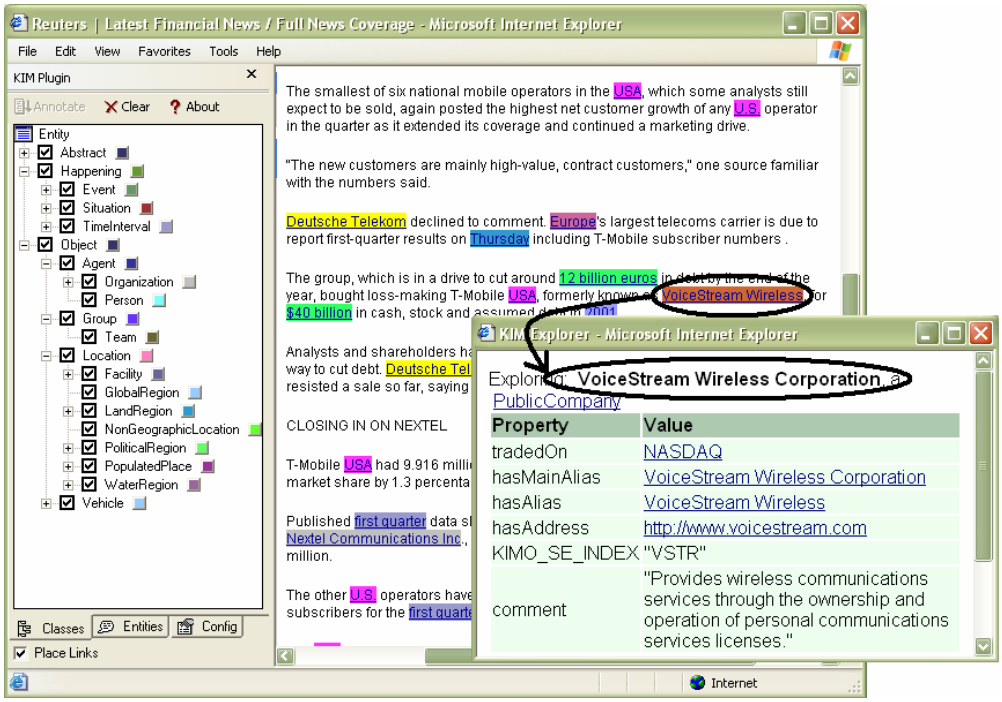


Fig. 3. KIM Plug-In, semantically annotated content and KB Explorer (on the front)

3.1. KIM Architecture

The KIM platform consists of KIM Ontology (KIMO) [8], knowledge base, KIM Server (with API for remote access, embedding, and integration), and front-ends (browser plug-in for Internet Explorer, KIM web user interface with various access methods, and Knowledge Explorer for KB navigation). The KIM API provides semantic annotation, indexing and retrieval services and infrastructure. KIM ontologies and knowledge bases are kept in semantic repositories based on cutting edge Semantic Web technology and standards, including RDF(S) repositories (SESAME[9] [5]), and ontology middleware[10] [15]. KIM provides a mature infrastructure for scalable and customizable information extraction, as well as annotation and document management, based on GATE [8]. The Lucene[11] information retrieval engine has been adapted to index documents by entity types and measure relevance according entities, along with tokens and stems. It is important to mention that KIM, as a software platform, is domain and task independent as are GATE, SESAME and Lucene. The KIM Architecture diagram is depicted on Fig. 4.

[8] http://www.ontotext.com/kim/2003/03/kimo.rdfs
[9] http://sesame.aidministrator.nl/, RDF(S) repository by Aidministrator b.v.
[10] OMM, http://www.ontotext.com/omm. Ontology Middleware Module is an enterprise back-end for formal knowledge management.
[11] Lucene, http://jakarta.apache.org/lucene/, high performance full text search engine

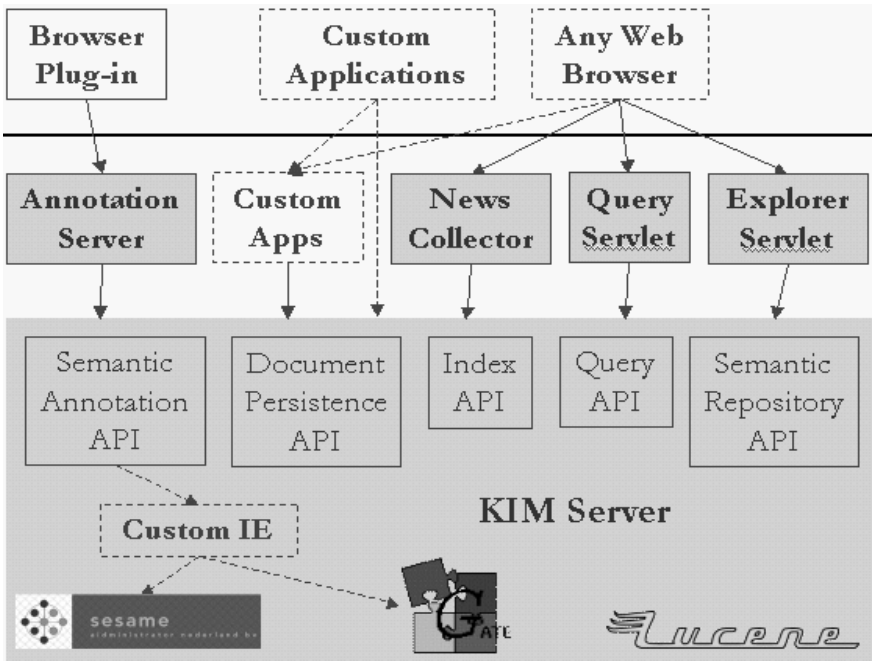


Fig. 4. KIM architecture – maybe change to include the KIM Web UI

3.2 KIM Ontology

The KIM ontology (KIMO) is a simplistic upper-level ontology starting with some basic philosophic distinctions between entity types (such as **`Object`**s - truly existing entities as locations and agents, **`Happening`**s - defining events and situations, and **`Abstract`**ions that are neither objects, neither happenings). Further on the ontology goes in more details, specifying real-world entity types of general importance (meetings, military conflicts, employment positions, commercial, government and other organizations, people, different types of locations, etc.). Also the characteristic attributes and relations for the featured entity types, are defined (e.g. **`subRegionOf`** property for **`Location`**s, **`hasPosition`** for **`Person`**s, **`locatedIn`** for organizations, etc.) Having this ontology as basis, one could add domain-specific extensions to it easily, to profile the semantic annotation for concrete applications. The integration of more than one domain-specific extension in a single application would not be possible without the intermediate role played by the upper-level ontology.

The KIM ontology (KIMO)[12] consists of 250 general entity types and 100 entity relations. The top classes are **`Entity`**, **`EntitySource`** and **`LexicalResource`**. The **`Entity`** class is further specialized into **`Object`**, **`Abstract`** and **`Happening`**. The top

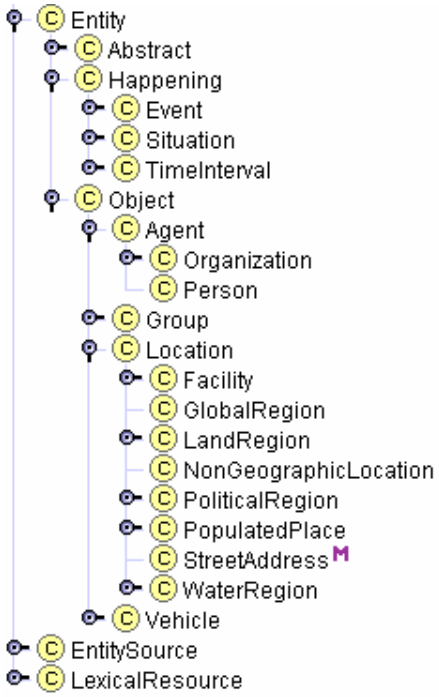
[12] http://www.ontotext.com/kim/2003/03/kimo.rdfs

Entities could be seen in the type hierarchy of the KIM plug-in on Fig. 3, and separately on Fig.5.
The **LexicalResource** branch is dedicated to encoding various data aiding the IE process, such as company suffixes (AG, Ltd.), person first names, etc. (depicted on Fig.5)

An important sub class of this branch is **Alias**, representing the alternative names for an **Entity** (see Fig. 7). The **hasAlias** relation is used to link an **Entity** to its alternative names. The official name of an entity is referred by the **hasMainAlias** property.

The instances of the **EntitySource** class are used to separate the trusted (pre-populated) information in the KB from the automatically extracted. This is indicated by the **generatedBy** property of the entity individuals.

The distribution of the most commonly referred entity types varies greatly from domain to domain (e.g. in a news corpus, the locations would be a much higher percentage from all entity annotations, than in an email corpus.) As researched in [18], despite the difference of type distributions, there are several general entity types that appear in all corpuses – **Person, Location, Organization, Money** (amount), **Dates**, etc. Further the ontology defines more specific entity types (e.g. **Mountain**, as a more specific type of **Location**.) The extent of specialization of the ontology is determined on the basis of research of the entity types in a corpus of general news (incl. political, sport, and financial, etc.)



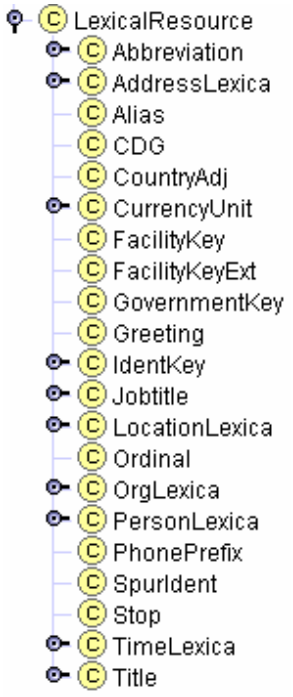


Fig. 5. The top of KIMO class hierarchy with expanded Entity branch. (on the left)
Fig. 6. The Lexical Resources top class hierarchy.

3.3 KIM Knowledge Base

The KIM KB represents a projection of the world, according to the domain that it is applied to. Our experiments are primarily in the field of international news. The specifics about this domain is that it covers the most well known and important entities in the world. KIM keeps the semantic descriptions of entities in the KIM KB, which is repeatedly enriched with recognized entities and relations. The entity descriptions are being stored in the same RDF(S) repository as the KIM ontology. Each entity has information about its specific type, aliases (incl. a main alias, expressing the (most probable) official name), attributes (e.g. **`Latitude`** of a **`Location`**), and relations (e.g. a **`Location subRegionOf Location`**). A simplistic schema of the entity representation is depicted on Fig. 7, where one could see the instance with its type and one alias.

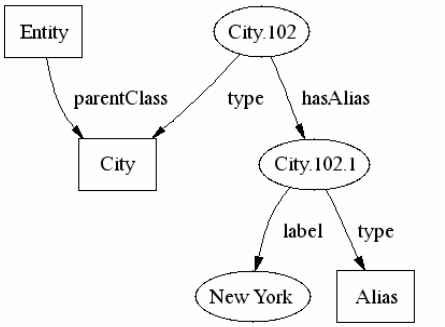


Fig. 7. Simplified view of the entity description

No matter how sophisticated the automatic IE process is, still one needs a starting KB to represent the entities that are considered important in the respective domain. This plenty of information should be carefully filtered in order to provide minimal, but representative coverage of the entities of general importance. There is no formal definition of the importance of an entity. However, we suggest that as important should be considered the entities that are well known to the wide public. Later on the importance of an entity could be represented through various ranking weights mostly derived statistically.

Pre-population of KIM KB.
KIM KB has been pre-populated with entities of general importance, that allow enough clues for the IE process to perform well on inter-domain web content. It consists of about 80,000 entities with more than 120,000 aliases. Various relations between entities are also predefined (like position of a person in an organization or company's allocation.)

The entities needed from the KB population are available on the web in the form of online encyclopedias, public servers, directories and gazetteers. For example the geographic locations and relations between them could be extracted [16] from NIMA's[13] GEOnet Names Server (GNS)[14], The Geographic Names Information System (GNIS)[15] data from the U.S. Geological Survey (UGCS), The Alexandria Digital Library (ADL) gazetteer[16], or other public geographic names server. The

[13] National Imagery and Mapping Agency of the US
[14] http://www.nima.mil/gns/html/
[15] http://geonames.usgs.gov/gnisform.html
[16] http://www.alexandria.ucsb.edu/

instances of important organization (and their officials) could be retrieved from the public directories of the biggest web portals, from other public servers, or in the form of compiled gazetteers.

KIM KB keeps the entity descriptions of frequently mentioned geographic resources. These entities have attributes and relations that depict their actual positioning and co-positioning in the physical world (such as **`longitude`**, **`latitude`**, **`subRegionOf`**). The GNS (GEOnet Names Server) has been used to extract the instances of the Location class. One of the most important relation types is **`subRegionOf`**, carrying the meaning that a region is a part of another one (e.g. **`Country subRegionOf Continent`**.) In its current state the KIM KB contains about 50,000 locations, including continents, global regions, 282 countries with their capitals, 4,700 cities (including all the cities with population over 100,000), mountains, big rivers, oceans, seas, and even oil fields. Each location has geographic coordinates and several aliases (usually including English, French and sometimes the local transcription of the location name) as well as co-positioning relations (e.g. **`subRegionOf`**.)

In the sources mentioned, the importance of the entities is not presented in an explicit form, and often there are even no clues for distinctions by this criterion (e.g. England and Scotland are listed in GNS alongside 40 other UK areas). On the other hand, some sources have inherent global importance specification of the contained entities (e.g. UN's list of cities with population over 100,000), but lack detailed attributes and relations, and cannot be used by themselves. The instances listed in such repositories are matched against exhaustive resources (e.g. GNS) and thus the significant entities are let through the filter, retaining their complementary disposition features (spatial attributes, and **`subRegionOf`** relations.)

The organizations with highest general importance also have been pre-populated in the KB. Including the biggest world organizations (such as UN, NATO, OPEC), over 7,900 companies, and 140 stock exchanges for a total of 8,400 organization instances. For the public companies (counting 5000 entities) there are 5500 position relations of managing personnel. The organizations also have locatedIn relations to the corresponding country instances. The additionally imported information about the companies consists of short description, URL, reference to an industry sector, reported sales, net income, and number of employees. The company data came from various sources, mostly per-country lists of registered companies. The company data is verified to contain all the publicly traded companies listed in the Google directory[17], Hoovers Online[18] and is currently being re-evaluated and enhanced with other important companies, according to the classifications of Forbes[19], Fortune magazines[20], and the European business directory[21].

Famous people (e.g. government officials, public company managing personnel) and some specific organizations (e.g. TV companies), have been also imported in the KB.

[17] http://directory.google.com/Top/Business/Major_Companies/Publicly_Traded/
[18] www.hoovers.com
[19] www.forbes.com
[20] www.fortune.com
[21] www.europages.net

In order to enable the IE process to recognize new entities and relations that are not a part of the KB, a collection of lexical resources is also presented in the KB. It covers organization suffixes, person names, time lexica, currency prefixes and others, serving as clues for the NER process.

Ensuring the quality of the KB content, is not trivial and could not be performed manually (having more than 80,000 pre-populated entities, the manual approach will simply not scale). The KIM KB is iteratively verified against an independently built KB of entities and relations collected manually from various web sources.

4 Semantic Information Extraction

The essence of the KIM IE approach is the recognition of named entities (NE) with respect to formal upper-level ontology (KIMO). The NE annotations are typed with respect to the entity classes in the ontology. The entity instances all bear unique identifiers that allow the annotations to be linked to the exact individual in the KB. The IE involved in KIM is currently concentrated mostly on the NER task, which is considered a step-stone for further attribute, relation, event, and scenario extraction. In order to identify the references of entity relations in the content, one should first have identified the entities. Usually the entity references are associated with a NE type, such as **`Location`**, **`Person`**, etc. More and more hierarchical NE type sets appear (f.e. [22]), especially for domain-specific applications. This is due to the need for finer grained specification and identification of world concepts. For example, it would be natural for an IE application performing company intelligence to keep more specialized sub-classes of **`Organization`** (e.g. such as **`PublicCompany`**). A NE type taxonomy however brings in a new level of complexity and (as discussed in section 5) sets new challenges for the evaluation of the performance, since the traditional Precision/Recall metrics are not directly applicable.

The IE process presented here uses light-weight ontology (KIMO) defining the entity types (called classes in the ontology slang.) In addition to the hierarchical ordering, each class is coupled with its appropriate attributes. The relation types are also defined with their domain and range restrictions. Actually, the basic ontology language used (RDFS) considers both the relations and attributes as properties, which can also be ordered in a hierarchy. Further, the ontology also has a branch of lexical resource classes (section 3.2). Given the ontology, the entities in the text could be linked to their type, which is also feasible with just a type taxonomy. However we would like to go further, and identify not only the type of the NE but also keep its semantic description and extend it with the IE process. Thus the NE references in the text are linked to an entity individual in the KB (section 3.3). The accessibility of the semantic descriptions of entities in the KB would allow the IE process to later base on attributes and relations as clues for recognition and disambiguation. For example, if a Person appears along with a Company in the content, and there are two companies that have the mentioned alias we have ambiguity. A possible approach would be to check whether the Person has some relations with one of the companies (e.g. working in it), and if so, the related Company to be chosen as a better candidate and associated with the NE reference in the content.

It is important to mention the opportunities that such IE would reveal for the access methods. Indexing (with customized Lucene) over the entity references in the text allows later on to perform IR with respect to entities. Thus one could specify the entities that are expected to be part of the result set of documents, with attribute and name restrictions (e.g. a **`Person`** which name ends with 'Alabama'). To solve this task we apply the semantic restrictions over the entities in the KB. Then the documents referring the resulting entities are being returned with ranking according to NEs. Even more one could specify a pattern of entities and relations between them, and restrict the entities by attributes, name and type.

KIM IE is based on the GATE framework, which has proved its maturity, extensibility and task independency for IE and other NL applications. We have reused much of GATE's document management functionality, and generic NLP components as its *Tokenizer, Part-of-Speech Tagger*, and *Sentence Splitter*. These processing layers are provided by the GATE platform, along with pattern-matching grammars, NE coreference and others, as standardized building bricks for easy construction of sophisticated IE applications.

For our purposes we changed the grammar components to handle entity class information and match rules according to it. The grammar rules are now based on the ontology classes, rather than on a flat set of NE types. This allows much more flexibility in the creation of NER rules at the most appropriate level of generality, giving both the opportunity to generalize and handle more specific NE types. A rule trying to extract relation between an organization and its point of presence can be specified at the level of the most general classes it applies to (**`Organization`** and **`Location`**) and still match a patterns with much more specific information (say, a radio station located in a county). On the other hand, instead of referring to all locations we could prefer to have rules that are especially applicable for **`Countries`**, **`Cities`**, or **`Seas`**.

The *Semantic Gazetteer* lookup component is based on the entities and lexical resources in the KB, rather than on file lists of aliases. Along with it all the reused components have been opened towards the semantic repository. For example the NE coreference module, in addition to the traditional ortho-matching techniques, handles the instance information of NE annotations and matches them according to it., as well as the traditional substring transformation matching. The *Semantic Gazetteer*, the simple disambiguation and annotation filtering components, as well as the final KB enrichment layer have been developed from scratch. These are not innate to a traditional NER and are inquired by the specifics of the Semantic IE, which takes care of the identification of NE references with respect to the ontology and KB.

The IE component flow diagram (Fig. 8) displays the sequential processing of content to the point where semantic annotations of NE are produced over it. The semantic repository is also displayed and linked with the ontology and KB aware components. The semantically-aware modules are presented in sub-sections below.

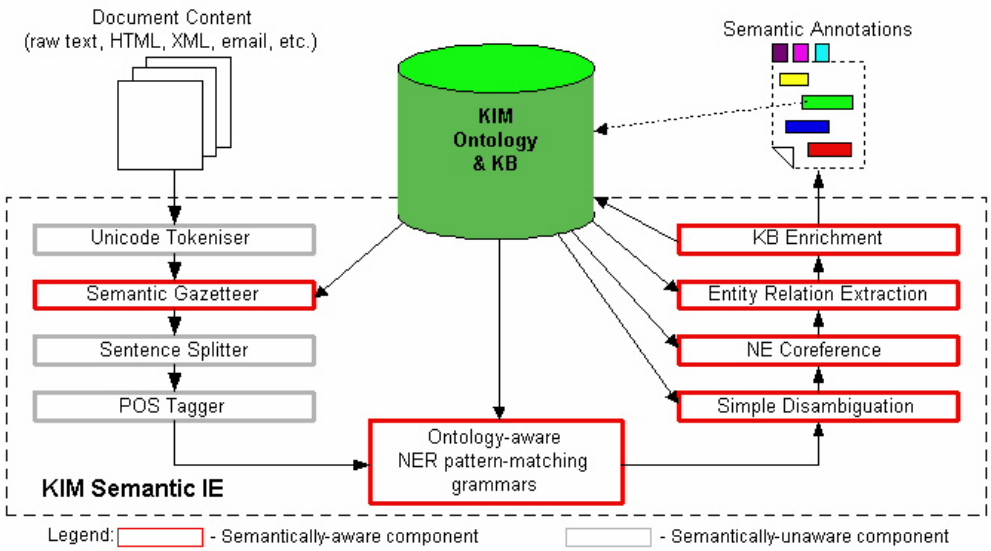


Fig. 8. KIM Semantic IE flow diagram.

4.1 Semantic Gazetteer

In the *Semantic Gazetteer* the lists of a traditional text-lookup component have been exchanged with a knowledge base that keeps the entities with their aliases and descriptions, as well as the lexical resources (such as possible male person first names). These are used to initialize the *Semantic Gazetteer* component, which keeps the various aliases and their type and instance references (as URIs). Upon occurrence of a known lexical resource or entity alias in the text (f.e. *Monday, John, GMT*, etc.), the *Semantic Gazetteer* generates a temporal annotation with a link to a class in the ontology (f.e. *Monday* will be linked to the KIM ontology class **`http://www.ontotext.com/kim/kimo.rdfs#DayOfWeek`**). Even more, the aliases of entities in the text are linked to the specific instances they refer to (f.e. *California* will be linked to the instance **`http://www.ontotext.com/kim/kimo.rdfs#Province.4188`**).

Since, many entities share aliases (f.e. *New York* is both a **`Province`** and a **`City`**) it often happens that one NE reference in the text is associated with several possible types and instances. At this phase we make sure all the equivalent possibilities are generated as annotations. Later on simple disambiguation techniques (section 4.4) are applied to filter some of the alternative annotations.

Although the KB contains both pre-populated and automatically recognized entities, only the former are used in the lookup process. The entities extracted from the processed content are not considered, and thus possible recognition mistakes are not reused as evidences. Let's consider we have previously extracted that within a given context the alias '*John*' referred to an entity with main alias (official name) '*John Smith*' and this entity with its semantic description has been added to the KB

and to the *Semantic Gazetteer* model. If the *Semantic Gazetteer* considered the recognized entities, the next time that '*John*' appears in the content it will be linked to the '*John Smith*' entity individual, and to many others with the same first name. But since the reference '*John*' doesn't really give a clue that one of the recognized entities with this first name is mentioned, the extracted information should be used with caution.

This phase is the entry-point for association of annotations in the text with a class in the ontology, and (for the entities) an instance in the KB. From here on the temporal annotations bare these semantic links, upon which the rest of the IE components base their processing.

4.2 Ontology-Aware Pattern-Matching Grammars

Pattern-matching grammars have proven to be applicable for various NLP tasks and also have traditionally been used for IE and NER. A grammar processor called JAPE [9] is a part of the GATE platform, and allows the specification of rules that fire on patterns of annotations. Thus one could specify actions and transformations that would take place if the rule is fired from a pattern in the content. We have modified the JAPE processor to handle class information and match patterns of annotations according to it. The NE grammars are based on the ones used in ANNIE[22] within the GATE project. In the modified grammars the definition of a rule goes through specification of the class restrictions for the entities in the pattern. The matching process uses the ontology to determine whether the candidate annotation has the same class as (or a sub-class of) the class in the pattern. Thus one could specify a pattern referring to a more general class (e.g. **`Organization`**), allowing all of its sub-classes (e.g. commercial, educational, religious and other organizations) to fire the grammar rule.

The pattern matching grammars are initially used to determine the entities within the processed content. At this point the suggested (by the *Semantic Gazetteer*) candidates for entities are evaluated. Some of them are considered credible and are transformed to final NE annotations. These inherit the type and instance information from the lookup annotations generated by the gazetteer. Other NE annotations are constructed by the grammar processor according to patterns in the content. These annotations have an entity type, but lack the instance information since they have not yet been associated with an existing KB individual. An example for identification of entities missing in the KB is using location/organization pre/post keys - "*River* Thames", "Mitsubishi *Corporation*", etc. Some context-based clues are also considered, such as "*in*" followed by Token-with-first-uppercase testifying that the latter is a **`Location`** (e.g. *in Kyoto*).

Later on, template relations extraction takes place, identifying some relations that the entities manifest in the content (determining the place where an organization is located; determining people's positions in organizations, f.e. *the CEO of NorthernStar, Mr. Yamamoto*).

[22] ANNIE, open-source, robust Information Extraction (IE) system based on pattern-matching grammars realized with finite state algorithms. http://gate.ac.uk/sale/tao/index.html#annie

4.3 Orthographic NE Coreference

The NER process continues with orthographic NE coreference component (see [2] and [11] for more on NE coreference within GATE), that generates lists of matching entity annotations within one type, according to their text representation (e.g. names like Mr. Malkovich and John Malkovich are usually referring the same entity individual within given context).

We have extended the coreference module to take into account the instance information of the recognized entities, thus enabling different string representations of an entity to be matched if they are aliases of one and the same KB individual. Without the instance data, names like Beijing and Pekin could not be matched only on the basis of orthographic coreference algorithms. The result of the coreference component is that groups of matching entities are identified. Later on these groups are used to determine the instance information and the aliases of new entities.

4.4 Simple Disambiguation

Potentially there are multiple entity-aliases in the KB that are equivalent to a NE reference in the text. For such references the *Semantic Gazetteer* generates multiple alternative annotations. Thus the over-generation of semantic annotations is rooted in the richness of the KB and the phenomena of naming different things with same names (e.g. Moscow being a **`CountryCapital`** and a **`City`** in US). At the level of the NER during the gazetteer lookup phase it is impossible to disambiguate because of the lack of clues (i.e. the gazetteer layer does not use evidence from other components, but the raw content itself). Later on simple disambiguation techniques take place in the pattern-matching grammars phase. For example, ambiguity between **`Person`** and **`Organization`** (e.g. "U.S. Navy"), would normally be recognized as a person name from the pattern: *two initials + first uppercas*, but in this case the initials match a location alias. Another problem is the occurrence of locations in person names, e.g."*Jack London*" (disambiguated because in the KB, "*Jack*" is known as a person first name).

Another class of ambiguities is the appearance of two annotations with different class and instance information over the same entity reference (*New York* being a **`Province`** and a **`City`**). Currently disambiguation of such annotations is not performed and this is subject of future work. For example, the context could be scanned for entities related to the ambiguous ones and thus relevance of the alternative entities to the content could be evaluated. For instance, if *Moscow* is used along with *Russia* its relevance is higher than the relevance of the alternative American city. We would also experiment with other approaches towards disambiguation of named-entity references. Adaptation of HMM learner, that has already successfully been used for non-semantic disambiguation is one of the first ideas. We would also like to experiment with techniques similar to those used for word-sense disambiguation (namely, lexical-chaining) and "symbolic" context management.

Beside the disambiguation in the grammar rules, a thin annotation filtering layer is used. More than one overlapping entity annotations (with same types) could be

recognized over the same part of the content. This is due to alternative patterns that fire the same rule or multiple trusted entities with the same alias. For example a person title (*Mr*.) followed by a looked up person candidate (e.g. *John Malkovich*), could match the left hand side of a rule, that also has an alternative firing pattern to match person titles followed by a token with upper-cased first letter (instead of looking for temporary person annotations as in the first pattern). As a result of the filtering only the annotations with distinct instance data are admitted - e.g. *New York* would be recognized both as a city and as a province, thus allowing later context-based disambiguation to determine the correct individual.

4.5 KB Enrichment

The last phase is not part of the standard IE systems, since it is related to the KB enrichment with new entity instances and relations. The newly recognized entity annotations lack instance information and are still not linked to the KB. However these entity annotations could represent entities that are in the recognized part of the KB. The first step is to match the entity annotations by their class information and string representation against the set of recognized entities. If a matching entity individual is found, the annotation acquires its instance identifier. Otherwise a new entity individual is constructed and added to the KB along with its aliases derived from the list of matching entities (if such).

At this point all generated named entity annotations are linked to the ontology (via their type information) and to the KB (via their specific instance). The relation annotations generated by the template relation extraction grammars, are used to generate the according entity relations in the KB (e.g. person's positions; spatial positioning information for organizations, etc.).

This finalizes the IE process, having as a result named entity annotations linked to their semantic descriptions in the KB.

5 Evaluation of KIM Named Entity Recognition

Along with the enrichment of the KB and the evolution of the IE process, we repeatedly evaluate the NER performance of KIM. This is needed to detect in early phases erroneous processing components or data. In order to test KIM NER most correctly it should be evaluated versus a corpus annotated with the specific type information. Such a metric however is not trivial and is subject of future work.

To measure the NER performance of KIM IE we have modified the GATE Corpus Benchmark Tool (CBT). CBT compares sets of annotations (key and response set) and calculates Precision, Recall, and F-measure. The metrics are presented separately for each document and combined for the final result. We also use the CBT to evaluate two sequential versions of the KIM platform against a human annotated corpus, thus determining the changes of the performance from version to version (regression testing).

The KIM NER performance has been evaluated, using CBT, against a corpus, human annotated with named entities. The evaluation corpus contains 100 documents

of news articles from UK media sources (Financial Times, Independent, and Guardian). The corpus is annotated with the traditional flat NE types used by most of the NER systems (**`Location`**, **`Organization`**, **`Person`**, **`Date`**, **`Percent`**, and **`Money`**). Despite the fact that KIM provides more specific type information, it is still possible to test it against the human annotated corpus (because something that is a **`Mountain`** is also a **`Location`**). In Table 1 we present the Precision, Recall and F-Measure of the automatically annotated corpus versus the human annotated one. These metrics are about the correctness of the KIM named entity recognition process in terms of general NE types, on the flat level of abstraction in standard NER systems.

Annotation Type	Precision	Recall	F-Measure
Date	0.92	0.83	0.87
Person	0.86	0.88	0.87
Organization	0.79	0.65	0.71
Location	0.87	0.92	0.90
Percent	1	1	1
Money	1	1	1
Total	0.86	0.84	0.85

Table 1. Evaluation of KIM NER wrt general NE types.

6 Related Work

Significant amount of research on IE has been performed in various projects related to GATE (see [17], [2], [7] [8] [9], [11], [18]). GATE provides tools such as tokenizers, part-of-speech taggers, gazetteer lookup components, pattern-matching grammars, coreference resolution tools and others that aid the construction of various NLP and especially IE applications. GATE is also a framework for content and annotation management. KIM's IE and content management is grounded in the GATE framework, and opens it towards Semantic Web knowledge representation and management technologies.

For some time now it has been obvious that the several general NE types used by the IE systems are not specific enough for many applications, that there are much more categories that matter. NE type hierarchies design has been discussed in [22].

Semantic annotation of documents with respect to ontology and entity knowledge base is discussed in [6] and [14] – although presenting interesting and ambitious approaches, these do not discuss usage of information extraction for automatic annotation. The focus of Annotea [14] is manual semantic annotation for authoring web content, while [6] targets the creation of a web-based open hypermedia linking service, backed by a conceptual model of document terminology.

Semantic annotation is used also in the S-CREAM project presented in [13] – the approach there is interesting with the heavy involvement of machine learning techniques for extraction of relations between the entities being annotated. Similar approach is taken also within the MnM project [21], where the semantic annotations

can be placed inline in the document content and refer to an ontology and KB server (WebOnto), accessible via standard API.

An interesting named entity indexing and question/answer system is presented in [19]. Here flat set of entity types is assigned to tokens and the annotations are incorporated in the content, in order to index by NE type later. Once indexed the content is queried via natural language questions, with NE tagging over the question used to determine the expected answer type (e.g. When have the United Nations been established; UN here would be tagged with _ORG, thus specifying that the expected answer type is organization.) This approach is also interesting because of its question/answer interface, allowing the users to specify their queries in NL sentences (with few limitations).

Experiments with the acquisition of spatial knowledge and its usage for IE have been described in [16].

Significant work on ontology and metadata infrastructure has been undertaken in the KAON project [3], which shares similarities with SESAME [5].

All the semantic annotation techniques referred above lack the usage of upper-level ontologies and critical mass of world knowledge to serve as a trusted and reusable basis for the automatic recognition and annotation, as in the approach presented in [1] and discussed here. Also the IE processes involved in related work do not link the NE reference in the text with a NE individual in the KB. Because of this unique feature the semantic description of the entity instance reveals its attributes, aliases, type, origin source, and relations with other individuals.

7 Conclusion and Future Work

We presented the Semantic IE approach embodied in the KIM Platform, with the involved technologies and resources.

Even linguistically simplistic, KIM platform provides a test bed and proofs number of hypothesis and design decisions:

- It's worth using almost-exhaustive entity knowledge (sort of super-gazetteers) for information extraction. The technology used (based on GATE) can manage the scale. Even without significant efforts on disambiguation, the precision drawbacks are acceptable for many applications;
- It is possible to adopt a traditional symbolic IE system to perform semantic annotations and thus provide its results in shape suitable for Semantic Web applications;
- A simple but efficient technique for entity-aware IR is demonstrated based on indexing over semantic annotations, which is an interesting example of IR engine taking benefit of the IE process.

The implementation is currently under development, so, preliminary results are reported. The evaluation work done until now does not provide enough evidence regarding the approach, technology, and resources being used. The major reason for this is that there are neither test data nor well developed metrics for semantic annotation and retrieval.

There are number of challenges for the Semantic IE which we will address in our future work:

- Develop (or adapt) evaluation metric which properly measures the performance of a semantic annotation system;
- Experiment different approaches towards disambiguation of NE references
- Make use of more advanced IE-techniques for identification of relations, analysis of events and situations, etc.
- The KIM Ontology and KB as well as the methodology and procedure for their sustainable maintenance and improvement will be subject of future research.

References

1. Bontcheva K., Kiryakov A., Cunningham H., Popov B., Dimitrov M.. *Semantic Web Enabled, Open Source Language Technology.* In proc. of EACL Workshop "Language Technology and the Semantic Web", NLPXML-2003, 13 April, 2003
2. Bontcheva K., Dimitrov M., Maynard D., Tablan V., Cunningham H., *Shallow Methods for Named Entity Coreference Resolution.* Chaînes de références et résolveurs d'anaphores, workshop TALN 2002, Nancy, France, 2002.
3. Bozsak E. et al. KAON - *Towards a large scale Semantic Web*, EC-Web 2002
4. Brickley D, Guha R.V., eds. *Resource Description Framework (RDF) Schemas,* W3C http://www.w3.org/TR/2000/CR-rdf-schema-20000327/
5. Broekstra J., Kampman A., and van Harmelen F. - Sesame: An architecture for storing and querying RDF data and schema information. In H. Lieberman D. Fensel, J. Hendler and W. Wahlster, editors, Semantics for the WWW. MIT Press, 2001
6. Carr L., Bechhofer S., Goble C., Hall W.. *Conceptual Linking: Ontology-based Open Hypermedia.* In The WWW10 Conference, Hong Kong, May, pp. 334-342.
7. Cunningham H., Information Extraction: a User Guide (revised version). Department of Computer Science, University of Sheffield, May, 1999
8. Cunningham H., Maynard D., Bontcheva K. and Tablan V., *GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications*. In Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics, 2002.
9. Cunningham, H. and Maynard D. and Tablan V., JAPE: a Java Annotation Patterns Engine (Second Edition). Technical report CS--00--10, Univ. of Sheffield, Department of Computer Science, 2000.
10. Dean M., Connolly D., van Harmelen, F., Hendler J., Horrocks I., McGuinness D., Patel-Schneider P., Stein L.A., *Web Ontology Language (OWL) Reference Version 1.0.* W3C Working Draft 12 Nov. 2002, http://www.w3.org/TR/2002/WD-owl-ref-20021112/
11. Dimitrov M., Bontcheva K., Cunningham H., Maynard D., *A Light-weight Approach to Coreference Resolution for Named Entities in Text.* Proceedings of the Fourth Discourse Anaphora and Anaphor Resolution Colloquium (DAARC) , Lisbon, September 2002.
12. Fensel D., *Ontology Language, v.2 (Welcome to OIL)* . Deliverable 2, On-To-Knowledge project, December 2001. http://www.ontoknowledge.org/downl/del2.pdf
13. Handschuh S., Staab St., Ciravegna F., *S-CREAM – Semi-automatic CREAtion of Metadata.*The 13th International Conference on Knowledge Engineering and Management (EKAW 2002), ed Gomez-Perez, A., Springer Verlag, 2002.

14. Kahan J., Koivunen M., Prud'Hommeaux E., Swick R.. *Annotea: An Open RDF Infrastructure for Shared Web Annotations*. In The WWW10 Conference, Hong Kong, May, pp. 623-632.
15. Kiryakov A., Simov K.Iv., Ognyanov D., *Ontology Middleware: Analysis and Design* Del. 38, On-To-Knowledge, March 2002. http://www.ontoknowledge.org/downl/del38.pdf
16. Manov D., Kiryakov A., Popov B., Bontcheva K., Maynard D., Cunningham H., *Experiments with geographic knowledge for information extraction,* NAACL-HLT 2003, Canada., Workshop on the Analysis of Geographic References, May 31 2003, Edmonton, Alberta.
17. Maynard D. ,Tablan D. ,Ursu C., Cunningham H., Wilks Y., Named Entity Recognition from Diverse Text Types. *Recent Advances in Natural Language Processing 2001 Conference*, Tzigov Chark, Bulgaria.
18. Maynard D., Tablan V., Bontcheva K., Cunningham H, and Wilks Y., *MUlti-Source Entity recognition – an Information Extraction System for Diverse Text Types*. Technical report CS--02--03, Univ. of Sheffield, Dep. of CS, 2003. http://gate.ac.uk/gate/doc/papers.html
19. Moldovan D., Mihalcea R.. *Document Indexing Using Named Entities.* In "Studies in Informatics and Control", Vol. 10, No. 1, March 2001.
20. van Ossenbruggen J., Hardman L., Rutledge L., *Hypermedia and the Semantic Web: A Research Agenda*. Journal of Digital information, volume 3 issue 1, May 2002.
21. Vargas-Vera M., Motta E., Domingue J., Lanzoni M., Stutt A. and Ciravegna F,, *MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup*, In Proc. Of EKAW 2002, ed Gomez-Perez, A., Springer Verlag, 2002.
22. Sekine S., Sudo K., Nobata Ch., *Extended Named Entity Hierarchy* (LREC 2002)
23. Pustejovsky J., Boguraev B., Verhagen, M., Buitelaar P., and Johnston M., *Semantic Indexing and Typed hyperlinking*. In Proceedings of the American Association for Artificial Intelligence Conference, Spring Symposium, NLP for WWW, 120-128. Stanford University, CA, 1997.

Making Explicit the Semantics Hidden in Schema Models

Bernardo Magnini, Luciano Serafini, and Manuela Speranza

ITC-irst Istituto per la Ricerca Scientifica e Tecnologica,
Via Sommarive 18 - Povo,
38050 Trento, Italy
{magnini, serafini, manspera}@itc.it

Abstract. Most of the data stored in the Semantic Web is organized in schema models, which can be represented as labeled graphs where labels are short natural language expressions. Examples of schema models include ER-schema automata, ontologies, taxonomies, and Web Directories. The semantics of schema models is not explicit but is hidden in their structures and labels. To obtain semantic interoperability we need to make their semantics explicit by taking into account both the interpretation of the labels and the structures described by the arcs. We propose a methodology for interpreting schema models on the basis of the taxonomic relations and the linguistic material they contain. We rely on a set of linguistic repositories, such as WordNet, and explore a number of crucial linguistic issues such as disambiguation of polysemous words, multiwords, and coordinations. The Web Directories of Google and Yahoo! have been chosen as an evaluation set. We show that there is a considerable amount of information to be made explicit and discuss the performance of an implementation of our analysis.

1 Introduction

Hierarchical classifications are taxonomic structures used to organize large amounts of documents. The most typical examples of hierarchical classifications are file systems, marketplace catalogs, and the directories of Web portals. Documents of a hierarchical classification can be of many different types, depending on the characteristics and uses of the hierarchy itself. In file systems, documents can be any kind of file (e.g. text files, images, applications, etc); in the directories of Web portals, documents are pointers to Web pages; in the marketplace, catalogs organize either product cards or service titles.

Hierarchical classifications are quite useful for document classification and retrieval. Users browse hierarchies of concepts and quickly access the documents associated with the different concepts. The content of a concept is typically described by a label, but it also depends on the concepts at higher levels in the hierarchy, even though the relations between concepts are usually not explictly labeled.

Hierarchical classifications are now widespread as knowledge repositories and the problem of their integration and interoperability is acquiring a high relevance from a scientific and commercial perspective. A typical application of hierarchical classification interoperability occurs when a set of companies want to exchange

products without sharing a common product catalog. In these cases the best solution is to find mappings between their catalogs [17]. In [5], we have proposed an algorithm that finds semantic relations between the nodes of different hierarchical classifications. This algorithm strongly relies on a linguistic analysis of the labels contained in the classifications. The main difference between this algorithm and other approaches to schema matching such as [2], [7] and [4] is that in order to interpret a node of a hierarchy we do not limit ourselves to a linguistic analysis of its labels. Instead, we extend this analysis by considering the *implicit information* deriving from the *context* where the node occurs, i.e., the structural relations with the other nodes of the hierarchy.

Indeed, one of the most evident peculiarities of hierarchical classifications (and in general of schemas) is that the meaning of a node depends not only on the label of the node, but also on the position of the node in the hierarchy. Indeed, like databases and ontologies, concept hierarchies are built on taxonomic relations between concepts, but such relations are implicit and have to be interpreted. Like plain texts, hierarchical classifications contain linguistic material, i.e. labels that can be analyzed with NLP techniques; the context provided for a label, however, is not a sentence or a paragraph, but a set of concepts placed at different levels. Consequently, the interpretation is performed in two steps: first, each individual concept is analyzed separately from the others and is associated with a basic logical form. Then, on the basis of its position in the hierarchy and of its relations with other nodes, we build a full logical form for each concept.

For instance we can have a hierarchical classification of documents about sports that contains a node labeled with *Sports Organizations*, and a second hierarchical classification on the same topic, containing a node labeled with *Organizations* which is a child of a node labeled with *Sport*. Clearly the semantics of the node "Sports Organizations" in the first hierarchy coincides with the semantics of the node "Organizations" in the second hierarchy . This equality however cannot be discovered by simply analyzing the labels. One has to discover that the arc connecting "Sports" and "Organizations" is a specification arc. This interpretation is very ambiguous and context dependent. Consider the example of a node labeled with *Schools* with a descendant node labeled *United States*, in this case the hierarchical relation between the two nodes has to be interpreted as a location relation. This interpretation is based on the semantics that is hidden in the labels and in the hierarchical structure.

The aim of this paper is to describe a method to analyze the implicit knowledge hidden in hierarchical classifications and to make it explicit in order to provide a correct interpretation of its concepts. In particular, we describe an algorithm that, given a concept hierarchy, returns an interpretation of each node of a hierarchy in terms of a logical formula of a description logic [3]. The ideal output of our algorithm for the concept hierarchy of Figure 1 is reported in Figure 2. This algorithm does not consider the documents classified under the nodes, so that it can be used in situations where such information is partially available or not available at all.

The paper is structured as follows. In Section 2 we provide a formal definition of concept hierarchy based on intuitions on how documents are classified by humans. In Section 3 we describe the analysis of the concepts performed without considering the hierarchical structure of the context. In Section 4 we describe the interpretation of the concepts based on the structural relations of the concept

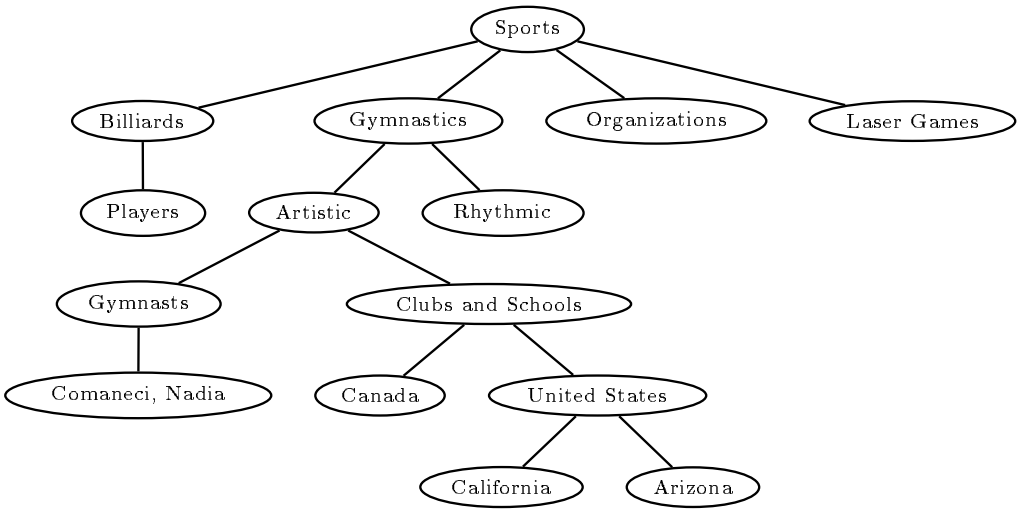


Fig. 1. Example of concept hierarchy (from Google Web Directories).

hierarchy. In Section 5 we describe the interpretation of implicit negations and disjunction. In Section 6 we discuss the results of an evaluation experiment where the procedure is applied to the Web Directories of Yahoo! and Google. Finally, Section 7 reports some relevant related work.

2 Concept Hierarchies

Here we introduce more formally the terms of our problem (see [18] for a more detailed description).

Definition 1 (Concept hierarchy) *We define a* concept hierarchy *as a triple* $H = \langle C, E, l \rangle$ *where* C *is a finite set of nodes,* E *is a set of arcs on* C*, such that* $\langle C, E \rangle$ *is a rooted tree, and* l *is a function from* C *to a set* L *of labels expressed in natural language.*

An example of concept hierarchy is provided in Figure 1, where a small part of the category 'Sport' in the Web Directories of Google is represented.

Definition 2 (Hierarchical classification) *A* hierarchical classification *of a set of documents* Δ *in a concept hierarchy* $H = \langle C, E, l \rangle$ *is a function* $\mu : C \rightarrow 2^{\Delta}$.

Classifications guide users in retrieving documents from the whole set Δ. The common procedure for seeking documents in a hierarchical classification is by entering the hierarchy from the root node, and, at each node, by choosing the child node under which the document is more likely to be classified. This choice is based on a semantic interpretation of the labels associated with the nodes, so that in most cases users do not need to check the content of the documents.

Consider, for instance, the concept hierarchy of Figure 1. In order to find documents about Romanian artistic gymnasts, a user would start from the root labeled with *Sports*, would first select *Gymnastics*, then *Artistic*, and then *Gymnasts*, and would finally retrieve the documents classified under this node. Users' choices are guided by the following facts:

- Understanding of the meaning of the labels attached to the nodes encountered during the navigation; in this case, *Sports, Billiards, Gymnastics, Organizations, Artistic, Rhythmic, Gymnasts* and *Clubs and Schools.*
- Knowledge of the fact that Romanian gymnasts are artistic gymnasts, and that gymnastics is a sport.
- Assumption that a document about artistic gymnasts is much more likely to be classified under the sub-tree rooted at *Sports/Gymnastics* than under the ones rooted at *Sports/Billiards* and *Sports/Organizations*. Similar assumptions are related to the choice between the children of *Sports/Gymnastics*, and so on.
- Awareness that the node *Gymnastics* is the most specific node about the topic 'Romanian gymnasts', as there is no node *Romania* available under *Gymnastics*.

In order to be useful for a user, a classification a should therefore respect a number of classification criteria, which can be summarized as follows:

M1 Each concept $c \in C$ has a *meaning* $m(c)$, which is some entity of a world domain.

M2 The meaning of a concept c depends only on the labels associated with a finite set of nodes $F(c) \subset C$ called the *focus of* c.

C1 A document $\delta \in \Delta$ is classified under a descendant node c' of c, i.e., $\delta \in \mu(c)$, only if δ is concerned with $m(c')$.

C2 A document δ is classified under the node c if c does not have any descendant c' such that δ is concerned with $m(c')$.

Criterion M1 guarantees that the hierarchical classification is done with a domain model in mind which is assumed to be shared by the users. Criterion M2 guarantees that the meaning of the concepts can be determined by visiting a finite (and possibly small) subset of the whole classification. Criteria C1 and C2 are standard classification criteria which can be found for instance in Yahoo! or in the Open Directory Project.[1] These two criteria provide the connection between the meaning of the labels and the set of documents classified under the node.

To formalize criteria M1-2 and C1-2, we provide a notion of *meaning* of a concept in a real world. In order to express meanings we adopt a logical approach, i.e. meaning are expressed in terms of formulas of a description logic

[1] Two foundamental criteria which are stated in many guidelines for Web Directories classification are the "Get specific" criterion and the "Look familiar" criterion.

Get Specific: When you add your document, get as specific as possible. Dig deep into the directory, looking for the appropriate sub-category. You can't submit your company to a top level category. [...] Dig deeper.

Look Familiar: Armed with the above knowledge, browse and search your way through the hierarchy looking for the appropriate category in which to add your company. Look for categories that list similar documents.

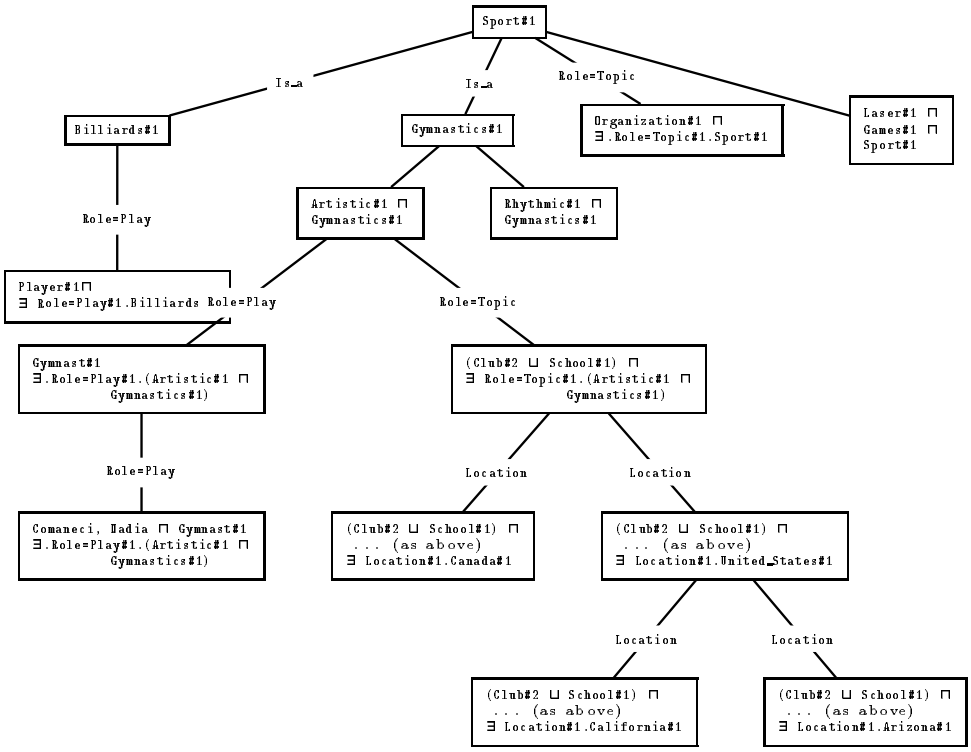


Fig. 2. The concept hierarchy of Figure 1 enriched with the meanings, and the relation types.

$\mathcal{DL}$, so that a label of a node can be associated either with a concept expression, or with a role description, or with an individual constant of $\mathcal{DL}$. The $\mathcal{DL}$ we adopt in this work is the description logics containing $\{\top, \bot, \sqcap, \sqcup, \neg, \exists R.$ and $\forall R.\}$. Furthermore, in the current work a concept c is associated with a meaning $m(c) = \phi$ which is a concept (roles and constants are not yet considered). For instance, the node *Sports* in Figure 1 corresponds to the primitive concept `sport`, while the node labeled with *Organizations* is associated with the concept $\texttt{Organization} \sqcap \exists \texttt{role} = \texttt{topic.sport}$, as it refers to sports organizations whose core business is sport.

This analysis is performed by using the semantic information provided in WordNet [8], which has been adopted because it is the largest repository of word senses and semantic relations currently available. The primitive concepts and roles of $\mathcal{DL}$ are selected among WordNet senses, on the basis of the labels occurring in $f(c)$. For instance, for the label *Sports*, WordNet provides 5 senses, which represent 5 different concepts of $\mathcal{DL}$. Again, for the word 'topic', WordNet provides different senses; for the role, we chose `topic#1`, meaning 'subject, theme'; another example of role is `location#1`, which describes spatial

relations between concepts. The process of finding the WordNet senses composing the label interpretation for a concept c is based on a linguistic analysis of $l(c)$. In order to minimize the ambiguity a further filtering of such senses is performed.

We build a basic label interpretation for the single concepts (see Section 3), and then we construct the contextual interpretation of every concept $c \in H$ by combining the basic interpretation of the concepts and their ancestors (see Section 4). The best we can obtain through the contextual interpretation of a concept hierarchy is an interpretation of each node such that the two criteria C1 and C2 are satisfied. For instance, the ideal output for the concept hierarchy in Figure 1 is reported in Figure 2.

3 Basic Label Interpretation

Concepts in concept hierarchies are described by labels, which in turn are composed by words and, possibly, separators between them. Labels are taken from a wide variety of linguistic expressions and can be single common words, such as *Dictionaries* and *Archaeology*, proper nouns, such as *Johann Sebastian Bach* and *California*, complex noun phrases, such as *Research Centers* and *Local Currency Systems*, prepositional phrases, such as *Sociology of Religion*, verb phrases, etc. More complex labels can also contain conjunctions, e.g. *Ecological and Environmental Anthropology*, punctuation, e.g. *Clubs, Teams, and Societies*, and acronyms, e.g., *GIS*.

In the first phase, i.e. basic label interpretation, we linguistically analyze the labels attached to the nodes and generate a formula in $\mathcal{DL}$ representing a first approximation of the meaning of the node.

Definition 3 (Basic label interpretation) *The* basic interpretation *is a function $lm : L \rightarrow \Phi_{\mathcal{DL}}$, that maps each single label $l \in L$ in a concept description $lm(l)$ of the description logic $\mathcal{DL}$ whose concepts are taken from the set of* WordNet *senses.*

Intuitively, a basic interpretation provides an interpretation of the concept label as a stand alone object. For instance, the basic interpretation of the node labeled with *Organizations*, occurring under *Sports*, would be equal to the basic interpretation of a node *Organizations* occurring under *Billiards*.

The first step of the procedure consists of text chunking, i.e. dividing each label into syntactically correlated parts of words. For this we run the Alembic chunker [6], developed by MITRE Corporation as part of the Alembic extraction system [1]. For example, with the label *Science Fiction and Horror*, the chunker first selects a part of speech for each word ('Science', 'Fiction', and 'Horror' are nouns, 'and' is a conjunction); then, it identifies two noun groups (NGs), i.e. 'Science Fiction' and 'Horror' (the syntactic head is marked in small capitals), and a coordinating conjunction between them (1a).

$$(1) \quad [(Science)_{nn}(\text{Fiction})_{nn}]_{NG}(and)_{cc}[(\text{Horror})_{nn}]_{NG}$$

The output of the chunker is used to transform each label into a basic logical form. A noun group consisting of more than one word is interpreted as the conjunction of the head and all its modifiers. For instance, $[(Science)_{jj}(\text{Fiction})_{nn}]$

is interpreted as [Science ⊓ Fiction], the reason being that the documents classified under a node with such label should be concerned both with 'science' and with 'fiction'.

The relations between different noun groups are interpreted on the basis of the linguistic material connecting them:

- coordinating conjunctions and commas are interpreted as a disjunction;
- prepositions, like 'in' or 'of', are interpreted as a conjunction;
- expressions denoting exclusion, like 'except' or 'but not', are interpreted as negations.

For example, *Science Fiction and Horror* is interpreted as a disjunction (2a), since under that node there might be both documents about 'science fiction' and documents about 'horror'; on the other hand, *Professional Photographers of America* and *Garments except Skirts* are examples of conjunction (2b) and negation (2c) respectively.

(2) *a*. [Science ⊓ Fiction] ⊔ [Horror]
b. [Professional ⊓ Photographers] ⊓ [America]
c. [Garments] ⊓ ¬ [Skirts]

The interpretation of proper nouns requires a process of named entities recognition (NER). The output of a chunker is passed to a rule-based NER system [13] which recognizes named entities and classifies them into one out of five categories (person, organization, location, measure, date). As an example, *J.S. Bach* is analyzed as in (3).

(3) <BNAMEX TYPE=PERSON *J.S.Bach* ENAMEX>

WordNet. In order to perform the semantic interpretation of the labels we access WordNet. We use a multilingual version of WordNet initially developed in the framework of the EuroWordNet Project [19] and currently in further development under the Meaning Project [16]. Five languages (English, Italian, Spanish, Catalan, and Basque) are aligned and additional semantic information, such as top ontology concepts, domains, selectional preferences, and distinctions between classes and instances, is provided. Moreover, we rely on the work carried on by [9], which aims at introducing formal distinctions in the WordNet framework. In particular, we make use of the following meta-level categories [10] associated with the synsets: TYPE, for synsets representing rigid properties (e.g. person#1), FORMAL ROLE, for synsets representing anti-rigid properties (e.g. student#1), and ATTRIBUTION, for synsets representing possible values of attributes (e.g. red#1, an attribute-value of color). We will use this meta-level information to construct appropriated logic forms for relations between concepts (see 'arc interpretation' in Section 4).

When a word is found in WordNet, all the senses of that word are selected and attached to the basic logical form. In the case of *Science Fiction and Horror*, for instance, WordNet provides all the three nouns contained in the label, and

so in the logical form we have the conjunction of the sets of senses of the three lemmas (4).

(4) [science* ⊓ fiction*] ⊔ [horror*]

We use the following notation: trade* denotes the disjunction

trade#1⊔trade#2⊔...⊔trade#n

of all the senses of 'trade' in WordNet; trade#3 indicates sense 3 of 'trade', while trade#[2,4] indicates the disjunction of senses 2 and sense 4.

Multiwords. When two or more words in a label are contained in WordNet as a single expression (i.e. a multiword), the corresponding senses are selected and, in the basic logical form, the intersection between the two words is substituted by the multiword. For instance, 'Science Fiction' is provided in WordNet as a single expression, so the logic interpretation is substituted by the senses of the multiword (5).

(5) [science_fiction*] ⊔ [horror*]

Word Sense Disambiguation. Since multiwords are much less polysemous than single words, the recognition of the multiwords provided in WordNet is a first step towards word sense disambiguation. A second step is performed by exploiting the relations between senses provided in WordNet. The label *maple tree*, for instance, is first transformed into [maple* ⊓ tree*], since 'maple' and 'tree' are polysemous words and 'maple tree' is not provided in WordNet as a multiword. However, maple#2 (defined as 'any tree or shrub of the genus Acer') is a second level hyponym of tree#1 (i.e. 'tree' as a woody plant), and so the meaning of 'tree' as a diagram (i.e. tree#2) can be discarded to obtain the disambiguated basic logical form [maple#2 ⊓ tree#1].

4 Contextual Interpretation

An interpretation of a concept in a hierarchical classification as a stand alone object, however, is partial, as the meaning of a node depends on the context where the node occurs (see criterion M2). Intuitively, the focus $f(c, H)$ of a node c belonging to H is the part of H that the user is required to visit in order to understand whether a document is in c. The contextual interpretation of a node c gives a meaning to the node on the basis of the meaning of the nodes belonging to its focus (i.e. the ancestors of c with their direct descendants).

Let $\mathcal{H}$ be the class of concept hierarchies, and $\mathcal{C}$ the class of nodes occurring in some $\mathcal{H}$.

Definition 4 (Contextual interpretation) *A* contextual interpretation *is a function* $m : \mathcal{C} \times \mathcal{H} \rightarrow \Phi_{\mathcal{DL}}$, *where* $\Phi_{\mathcal{DL}}$ *is a concept of a description logics* $\mathcal{DL}$, *such that* $f(c, H) = f(c', H')$ *implies that* $m(c, H) = m(c', H')$.

Once we have associated a contextual meaning $m(c)$ to a concept c, we can define the class of documents classified under c to be the set $\Delta(c)$ such that for each $\delta \in \Delta(c)$ containing the document δ satisfying the following conditions:

C1: One of the main topics of δ is a concept ϕ, and ϕ subsumes $m(c)$, i.e., $\phi \sqsubseteq m(c)$;
C2: For any descendant c' of c, ϕ does not subsume $m(c')$, i.e., $\phi \not\sqsubseteq m(c)$.

Given the concept hierarchy H, the main task described in this Section is to find a proper contextual interpretation $m(c, f(c))$ by combining the linguistic analysis of the labels associated with c and $f(c)$, with the information provided by the structure of $f(c)$. In the following we describe how we deal with word sense ambiguity, with multiwords and arcs interpretation, taking advantage of the context provided by the hierarchical classification.

Multiwords in context. The recognition of multiwords can also be performed on different contiguous levels. For instance, in WordNet there is a multiword 'billiard player', so in a hierarchy where *Players* has *Billiards* as a parent node, its basic logic form is substituted by the senses of the multiword.

Word Sense Disambiguation in context. The context of a concept is taken into consideration to perform further disambiguation of the concept itself. We perform word sense disambiguation by taking into consideration both structural relations between labels and conceptual relations between words belonging to different labels.

Let L be a generic label and L^1 either an ancestor label or a descendant label of L and let $\mathtt{s*}$ and $\mathtt{s}^1\mathtt{*}$ be respectively the sets of WordNet senses of a word in L and a word in L^1. If one of the senses belonging to $\mathtt{s*}$ is either a synonym, a hypernym, a holonym, a hyponym or a meronym of one of the senses belonging to $\mathtt{s}^1\mathtt{*}$, these two senses are retained and all other senses are discarded.

As an example, imagine *Apple* (which can denote either a fruit or a tree) and *Food* as its ancestor; since there exists a hyponymy relation between `apple#1` (denoting a fruit) and `food#1`, we retain `apple#1` and discard `apple#2`.

Arc interpretation. The intuition underlying the methodology we propose for arc interpretation is that it depends on the ontological features of the concepts or instances connected. Arcs connecting two nodes admit different interpretations on the basis of the meta-level categories (i.e. TYPE, FORMAL ROLE, ATTRIBUTION, and INSTANCE) of such nodes. Table 1 defines the description logics interpretation for all the possible combinations of the meta-level categories we have used.

According to these rules the arc connecting `gymnastics#1` (belonging to the category TYPE) and `artistic#1` (an ATTRIBUTE), for example, is interpreted as a ROLE relation (6a). In the case of `sport#1` and `gymnastics#1`, which are both types, we base the interpretation of the arc on WordNet; since a hyponymy relation between the two concepts is provided in WordNet, we interpret the arc as the intersection of the two concepts (6b).

(6) *a.* $\mathtt{gymnastics\#1} \exists ROLE.\mathtt{artistic\#1}$
b. $\mathtt{sport\#1} \sqcap \mathtt{gymnastics\#1}$

ARC	Description Logics Interpretation	Examples
$T_1 \rightarrow T_2$	a WordNet relation between T_2 and T_1, if available; $T_2 \sqcap T_1$, otherwise	Gymnastics $\rightarrow$ Sports
$T \rightarrow R$	$R \sqcap \exists$ ROLE.T	Organ $\rightarrow$ Organists
$T \rightarrow I$	$I \sqcap \exists$ ROLE.T	Cantatas $\rightarrow$ Bach
$T \rightarrow A$	$A \sqcap \exists$ ROLE.T	Clubs and Schools $\rightarrow$ Artistic
$R \rightarrow T$	$T \sqcap \exists$ ROLE.R	Organists $\rightarrow$ Organ
$R_1 \rightarrow R_2$	a WordNet relation between R_2 and R_1, if available; $R_2 \sqcap R_1$, otherwise	Composers $\rightarrow$ Artists
$R \rightarrow I$	$I \sqcap \exists$ ROLE.R	Organist $\rightarrow$ Bach
$R \rightarrow A$	$A \sqcap \exists$ ROLE.R	Gymnasts $\rightarrow$ Artistic
$I \rightarrow T$	$T \sqcap \exists$ ROLE.I	Canada $\rightarrow$ Clubs and Schools
$I \rightarrow R$	$I \sqcap R$	Comaneci, Nadia $\rightarrow$ Gymnasts
$I_1 \rightarrow I_2$	a WordNet relation between I_2 and I_1, if available; $I_2 \sqcap I_1$, otherwise	California $\rightarrow$ United States
$I \rightarrow A$	$A \sqcap \exists$ ROLE.I	Bach $\rightarrow$ Famous
$A \rightarrow T$	$T \sqcap \exists$ ROLE.A	Artistic $\rightarrow$ Gymnastics
$A \rightarrow R$	$R \sqcap \exists$ ROLE.A	Famous $\rightarrow$ Players
$A \rightarrow I$	$I \sqcap \exists$ ROLE.A	Young $\rightarrow$ Bach
$A_1 \rightarrow A_2$	$A_2 \exists$ ROLE.A_1	International $\rightarrow$ Rhythmic

Table 1. Interpretation of the arcs on the basis of the metalevel ontological categories of the concepts connected by the arcs, where the abbreviations 'T', 'R', 'I', and 'A' stand for TYPE, FORMAL ROLE, INSTANCE, and ATTRIBUTE respectively, and where the arrow means 'classified under' ('I $\rightarrow$ T' represents the arc between a TYPE and the INSTANCE classified under that TYPE).

Finally, in order to build the contextual interpretation of the nodes in a hierarchical classification, we combine the interpretation of the labels with the interpretation of the arcs. For example, the different contextual interpretations of `gymnastics#1`, `artistic#1`, and `gymnasts#1` are represented in 7a, 7b, and 7c respectively.

(7) *a*. $\texttt{sport\#1} \sqcap \texttt{gymnastics\#1}$
b. $\texttt{sport\#1} \sqcap (\texttt{gymnastics\#1} \sqcap \exists ROLE.\texttt{artistic\#1})$
c. $\texttt{sport\#1} \sqcap (\texttt{gymnastics\#1} \sqcap \exists ROLE.\texttt{artistic\#1}) \sqcap \exists ROLE.\texttt{gymnast\#1}$

5 Implicit Disjunctions and Negations

Implicit Disjunctions. As explained in Section 3, the presence of a coordinating conjunction makes the disjunction between noun groups within a label explicit , but we can also have implicit disjunction between elements placed at different levels of the hierarchy (concepts with a disjoint descendant).

As an example, let's take a concept hierarchy with the root *Soccer*, a descendant *Leagues*, and a further descendant *Clubs*, which admits two conflicting interpretations: from the point of view of the hierarchical structure, *clubs* denotes a subset of *leagues* (being a child of it); on the other hand, from the point of view of the world knowledge provided in WORDNET, [`club#2`] (defined as 'a formal

association of people with similar interests') and [`league#1`] (defined as 'an association of sports teams'), can be considered as disjoint because they have the same hypernym, i.e. `association#1`. In order to combine the two information sources, *Leagues* has to be reinterpreted as if it were *Leagues and Clubs* (8a).

$$(8) \quad [\texttt{soccer}*] \sqcap [[\textbf{league\#1}] \sqcup [\textbf{club\#2}]]$$

When two concepts in a path are disjoint, i.e. when a concept is disjoint from a concept that is either an ancestor or a descendant, the meaning of the ancestor has to be reinterpreted. More formally:

Let c and c' be two concepts, and let `c#i` and `c'#j` be two senses of c and c' respectively. We apply the following rule:

- replace the sense `c#i` with `c#i` *and* `c'#j`, if `c'#j` is disjoint from `c#i` and c is an ancestor of c'[2]

Implicit Negations. Similarly, the negation can be marked by expressions like 'but not' or 'except', but can also be implicit in the case of elements belonging to different labels (inclusion relation between two siblings). For instance, in Google Web Directories we have *Sociology* and *Science* as sibling nodes classified under *Academic Study of Soccer*; from the point of view of world knowledge, sociology is a science (and in fact in WordNet `sociology#1` is a second level hyponym of `science#2`). As a consequence, the node labeled with *Science* has to be interpreted as if it were *Science except Sociology*.

Whenever a concept in a label has a part-of relation or an is-a relation with a concept in another label on the same level, it is necessary to re-interpret the meaning of the more general concept. More formally:

Let c and c' be two concepts, and let `c#i` and `c'#j` be two senses of c and c' respectively. We apply the following rule:

- replace the sense `c#i` with `c#i` $-$ `c'#j`, if `c#i` is either a hyponym or a meronym of `c'#j` and c and c' are siblings

6 Experiments

As a test set for the evaluation of the algorithm for the interpretation of concept hierarchies, which has been implemented in Java, we have focused on the Web Directories of Yahoo! and Google, where documents are represented by millions of Web page URLs'.

Yahoo! and Google Web Directories have respectively fourteen and fifteen main categories (e.g. 'Computer & Internet', 'News & Media', 'Recreation & Sport', 'Health', 'Society & Culture', 'Arts & Humanities', 'Science', 'Social Science' in Yahoo! and 'Arts', 'Computer', 'Health', 'News', 'Recreation', 'Science' and 'Society' in Google) consisting of a number of nodes ranging between a few thousand and tens of thousands nodes each. Each of these categories can be considered as the root of a sub-hierarchy.

[2] `s#k` is disjoint from `t#h` if `s#k` belongs to the set of opposite meanings of `t#h` (if `s#k` and `t#h` are adjectives) or, in the case of nouns, if `s#k` and `t#h` are different hyponyms of the same synset.

	Yahoo! Arch.	Google Arch.	Yahoo! Med.	Google Med.
# Concepts	105	312	703	1,023
Average label repetition	1.0	1.3	2	1.8
# Words	170	521	1,231	1,549
# Words/label	1.6	1.7	1.8	1.5
WordNet's coverage	95.5%	91.5%	88.7%	91.4%
Average polysemy	3.8	3.7	4.6	3.2
# Multiwords	11	45	51	116
# Disjunctions	10	51	99	58
# Conjunctions	41	109	239	325

Table 2. Analysis of the Architecture and Medicine sub-directories in Yahoo! and Google.

A preliminary analysis has been performed on two sub-hierarchies, i.e. 'Architecture' (under the main category 'Art') and 'Medicine' (under the main category 'Health'), whose sizes range between one hundred and one thousand nodes (see Table 2). The labels attached to the nodes are generally short, with an average of 1.5-1.8 words per label. Labels can be repeated more than once since the same label can be attached to different nodes in different places of the hierarchy; in fact, the bigger a hierarchy is, the higher is the average repetition of the labels (there are no repetitions in Yahoo! 'Architecture', while in Yahoo! 'Medicine' a label is repeated on average two times).

These two sub-hierarchies have been chosen among those where WordNet's coverage was highest (in fact, between 88.7% and 95.5% of the words and lemmas occurring in the labels are found in WordNet). It has been found that each lemma has on average between 3.2 and 4.6 senses, which makes the need for word sense disambiguation very important.

A manual and partial evaluation of the disambiguation process (we have checked manually the WordNet senses suggested by the algorithm for the 312 labels in Google 'Architecture') has shown that the procedure is very precise, with a precision rate ranging between 69 and 75%, but has a low recall, which is due to the fact that WordNet contains only hyponymy and meronymy relations and no other kinds of relations, like role or location relations.

As for the presence of multiwords, between 11.3% and 14.4% of the labels contain a multiword, which is remarkable, if we consider that between 54.8% and 62.6% of them consist of one single word. The good recognition of multiwords contributes to reduce the polysemy of concepts as most of them (almost 80% of the multiwords recognized in the experiment) are monosemous, with an average polysemy rate around 1.2 senses per multiword.

As far as negations are concerned, the hierarchies under analysis do not contain expressions denoting exclusion, while many implicit negations have been discovered. Our procedure works quite well with nouns denoting concrete objects like buildings. For example, in Google 'architecture', the node *Architecture/History/Periods and styles/Romanesque* has two descendants, *Churches* and *Cathedrals*; since cathedrals are actually churches (and an hyponymy relation between them is provided in WordNet), *Churches* is reinterpreted as if it were

Churches except Cathedrals. On the other hand, performance is not so good with abstract nouns, like states and events.

A limitation of the current system is that the use of the Alembic chunker does not permit the resolution of coordination ambiguities involving nominal compounds (and neither would the use of a more sophisticated parser, as this problem has received relatively little attention [15], when compared to other aspects, like for instance prepositional phrase attachment). To make an example, noun phrase coordinations with the form *n1 and n2 n3* admit two structural analysis, one in which *n1* and *n3* are the two syntactic heads being conjoined (9a) and one in which the conjunction is between the modifiers *n1* and *n2* (9b).

(9) *a* $(Nightclubs)\ and\ (Dance\ Halls)$
b $(Food\ and\ Drug)\ Administration$

Most of the times Alembic suggests the first analysis, which is correct in the case of *Nightclubs and Dance Halls* (10a), but is incorrect in many other cases, like *Food and Drug Administration* (10b) , which should be analyzed as in 10c. We plan to refine our analysis of coordinations involving nominal compounds by introducing rules based on number agreement as in [15].

(10) *a* $[(\textsc{Nightclubs})_{nn}]_{NG}(and)_{cc}[(Dance_{nn}(\textsc{Halls})_{nn}]_{NG}$
b $*[(\textsc{Food})_{nn}]_{NG}(and)_{cc}[(Drug)_{nn}(\textsc{Administration})_{nn}]_{NG}$
c $[((Food)_{nn}(and)_{cc}(Drug)_{nn})(\textsc{Administration})_{nn}]_{NG}$

7 Related Work

Our use of linguistic analysis to enrich hierarchical classifications with semantic information is related to the work presented in [20] on conceptual indexing. In that work the conceptual structure of phrases is analyzed using semantic relationships between words to establish connections between the terms in a conceptual taxonomy. Even if similar methodologies are applied, our approach aims at interpreting already existing taxonomies in order to make explicit a number of semantic relations, while in conceptual indexing the starting point is the extraction of terminology from documents.

WordNet is also used in [14] to give semantic interpretation to complex terms that have been automatically extracted from texts; relations between synsets are then exploited in order to organize concepts into trees. In our approach, on the other hand, the hierarchies represent the starting point, and structural information is used together with semantic information in order to interpret the labeled nodes.

Contextual interpretation of Web Directories headings has also been suggested in [12], but with a different aim. Here the knowledge embedded in the structure of the Directories is used to obtain labeled training data for Information Extraction from Web documents with limited human effort, while we aim at interpreting the Web Directories headings in order to discover the content of the classified documents without looking them up.

The problem of allowing for the interoperability of concept hierarchies, and in particular of catalogs, has been addressed also in [2]. Their approach is based

on the use of document classification algorithms; our methodology, on the other hand, does not use the documents associated with the nodes of the conceptual hierarchy, since it is based on the interpretation of labels and of the relations between them.

Finally, research related to the linguistic analysis of multi-word expressions and terminology has been conducted by Jacquemin and Morin in [11], who describe a framework for organizing multi-word candidate terms with the help of automatically acquired links between single-word terms derived from WordNet.

8 Conclusions

We have provided a formal semantics for hierarchical classifications and then used that formal framework to explore a number of linguistic issues crucial for interpreting the knowledge implicitly represented in such classifications.

The methodology we have proposed, based on a linguistic interpretation of the labels provided in the hierarchy, takes as input a concept hierarchy and returns the interpretation of each label. The process of interpreting a label coincides with the progressive construction of a logical form in description logics, where predicates are WordNet senses. It is performed in two steps: on the basis of the output of the chunker, basic logic forms are first built for each single concept independently of the others; then, full logical forms are built by combining the basic logic form of each concept with the basic logical forms of the nodes belonging to its focus.

In the future we plan to work on a systematic analysis of the performance of the methods with respect to the different steps and on the realization of a module for the discovery of the different kinds of relations between concepts, such as role, location, etc.

References

1. Aberdeen, J., Burger, J., Day, D., Hirschman, L., Robinson, P. and Vilain, M.: MITRE: Description of the Alembic System as Used for MUC-6. Proc. of the Sixth Message Understanding Conference (MUC-6), Columbia, Maryland, November, 1995
2. Agrawal, R. and Srikant, R.: On Integrating Catalogs. Proc. of the Tenth International World Wide Web Conference (WWW-2001), Hong Kong, China, May, 2001
3. Baader, F. and Nutt, W.: Description Logic Handbook. Pages 47-100, Cambridge University Press.
4. Bergamaschi, S., Guerra, F. and Vincini, M.: Product Classification Integration for E-Commerce. Proc. of WEBH-2002, Second International Workshop on Electronic Business Hubs. Aix En Provence, France. September, 2002
5. Bouquet, P., Magnini, B., Serafini, L. and Zanobini, S.: A SAT–based Algorithm for Context Matching. To appear in: Proc. of the Fourth International and Interdisciplinary Conference on Modeling and Using Context
6. Day, D.S. and Vilain, M.B.: Phrase Parsing with Rule Sequence Processors: an Application to the Shared CoNLL Task. Proc. of CoNLL-2000 and LLL-2000. Lisbon, Portugal, September, 2000

7. Doan, A., Madhavan, J., Domingos, P. and Halevy, A.: Learning to Map between Ontologies on the Semantic Web. Proc. of WWW-2002, 11th International World Wide Web Conference, Honolulu, Hawaii, May, 2002
8. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. The MIT Press, Cambridge, US, 1998
9. Gangemi, A., Guarino, N. and Oltramari, A.: Restructuring WordNet's Top-Level: The OntoClean Approach. Proc. of ONTOLEX 2002 (Workshop held in conjunction with LREC 2002), Las Palmas, Canary Islands, Spain, May, 2002
10. Guarino, N.: Some Ontological Principles for Designing Upper Level Lexical Resources. Proc. of LREC 1998, Granada, Spain, 1998
11. Jacquemin, E. and Morin, E.: Projecting Corpus-Based Semantic Links on a Thesaurus. Proc. of the 37th Annual Meeting of the Association for Computational Linguistics, University of Maryland, June, 1999
12. Kavalec, M. and Svatek, V.: Information Extraction and Ontology Learning Guided by Web Directory. Proc. of OLT-02 (Workshop on ML and NLP for Ontology Engineering) held in conjunction with ECAI 2002, Lyon, France, July 2002.
13. Magnini, B., Negri, M., Prevete, R. and Tanev, H.: A WordNet-Based Approach to Named Entities Recognition. Proc. of the Workshop SemaNet'02: Building and Using Semantic Networks, at COLING-02, Taipei, Taiwan, 2002
14. Missikoff, M., Navigli, R. and Velardi P.: Integrated Approach for Web Ontology Learning and Engineering. IEEE Computer, November, 2002.
15. Resnik, P.: Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. Journal of Artificial Intelligence Research 11, 2001
16. Rigau, P., Magnini, B., Agirre, E., Vossen, P. and Carrol, J.: MEANING: a Roadmap to Knowledge Technologies. Proc. of the workshop "A Roadmap for Computational Linguistics", COLING-02, Taipei, Taiwan, 2002
17. Schulten, E., Akkermans, H., Botquin, G., Drr, M., Guarino, N., Lopes, N. and Sadeh, N.: Call for Participants: The E-Commerce Product Classification Challenge. IEEE Intelligent Systems, 16-4, 2001
18. Serafini, L., Bouquet, P. and Donà, A.: CTXML Context Markup Language. Technical report, ITC-irst, 2002
19. Vossen, P.(Ed.): Special Issue on EuroWordNet, Computers and Humanities, 32, 1998
20. Woods, W.A.: Conceptual Indexing: A Better Way to Organize Knowledge. SUN Technical Report TR-97-61, 1997

A Natural Language Mediation System for E-Commerce applications: an ontology-based approach

Johannes Heinecke[1] and Farouk Toumani[2]

[1] France Télécom R&D, DMI/GRI/LN, F-22307 Lannion cedex, France,
johannes.heinecke@rd.francetelecom.com
[2] LIMOS, Université de Clermont-Ferrand, F-63173 Clermont-Ferrand, France,
ftoumani@isima.fr

Abstract. This paper describes how ontologies are used to mediate between languages and to infer answers to user questions in the multilingual e-commerce mediation system MKBEEM.[3] As an example, the paper discusses how a complex user request in human language is transformed into an ontological formula and subsequently exploited to identify a service which matches best. The MKBEEM-system prototype is in principle language independent and has been tested for the time being in Finnish, French, English and Spanish.

1 Introduction

The MKBEEM-project integrates knowledge-based processing (Knowledge Representation and Reasoning) and Human Language processing in providing multilingual e-commerce mediation services in order to allow a customer to use her own language, independent of the country where the product/service provider is based in. The consortium aims at proving that the technology concept is robust for given domains, and thereby bringing advances in both technology and services.

The global aim of the MKBEEM-project is to extend current electronic commerce platforms to reach a truly pan European and culturally open electronic commerce market. The main technical aim of MKBEEM is to create an *intelligent knowledge based multilingual* mediation service which displays the following features:

– Natural language interfaces for both the system's content providers/service providers and the end user.

[3] The project MKBEEM (Multilingual Knowledge Based European Electronic Marketplace, 2000-2002, http://www.mkbeem.com/) is a project funded by the European Commision (IST-1999-10589). The consortium, coordinated by France Telecom R&D (F), consists of VTT Information Technology (FIN), Universidad Politécnica de Madrid (E), National Technical University of Athens (GR), CNRS-LIRMM (F), SchlumbergerSema (E), Société Nationale des Chemins de Fer (F), and Ellos (FIN).

– Automatic multilingual cataloguing of products by service providers.
– On-line e-commerce contractual negotiation mechanisms in the language of the user, which guarantee safety and freedom.

Ontologies have been widely recognized as a central solution for sharing conceptions of goods and services among parties in e-commerce [1,2,3]. A recent survey by IBM and Icon Medialab found that in the Scandinavian countries on average 35% and in Finland up to 60% of purchase attempts failed in eShops. A major cause for this bad usability was that the customers could not find the requested products. Simple string based product search facilities are not enough. "No product available" is an insufficient answer, if the selection includes comparable goods or if the user just happens to use terms that differ from the ones in the catalogue. eShops need to solve the best possible offerings matching the user requirements, like human shopkeepers would do. The required question-answering capabilities can be realized by inferring based on domain ontologies, e.g. product models, and related generic ontologies. Moreover, ontologies can be used to facilitate multilinguality. In the MKBEEM-project, ontologies serve as the central solution for providing multilinguality and intelligent question answering [4]. The main result of this project is a multilingual e-commerce mediation system. It supports three main functionalities:

– *Multilingual cataloguing*, which enables providers to describe in their own language the goods and services that are on sale. Textual descriptions are translated automatically. Facts about products are extracted automatically into a language neutral form that complies with the product models of the domain ontology.
– *Processing of customer language information requests*, which is based on the co-operation between *human language processing* and *ontologies* of the commerce domain, the related products and generic common sense issues. Ontologies bridge between languages and also help in implementing fuzzy information search.
– *Multilingual trading*, which among other things applies an e-commerce ontology in carrying out contract terms adaptation for a particular shopping basket taking into account the countries of the seller and the buyer.

The MKBEEM-system prototype supports currently Finnish, French, English and Spanish. The technology can be easily adapted to other languages as well since all ontological knowledge is language independent. Feasibility tests have been conducted with test users since September 2002 in France and in Finland for mediating clothes, railway tickets, Finnish holiday cottages and French hotel room reservations, and car rental.

2 Technical Approach

In MKBEEM, ontologies are used to provide a consensual representation of the electronic commerce field in two typical domains (tourism and mail order) allowing the exchanges independently of the language of the end user, the service,

or the content provider [2,5]. Ontologies are used for classifying and indexing catalogues, for filtering user queries, for facilitating man-machine dialogues between users and software agents, and for inferring information that is relevant to the user requests.

The MKBEEM-ontologies are structured in three layers, as shown in Figure 1.

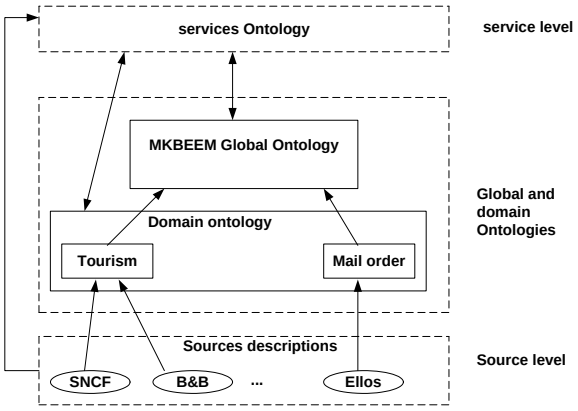


Fig. 1. Knowledge representation in the MKBEEM-system

The *global ontology* describes the common terms used in the whole MKBEEM-platform. This ontology represents the general knowledge in different domains (e.g., date, time) while each *domain ontology* contains specific concepts (e.g., trip) corresponding to vertical domains such as tourism and mail orders. The *service ontology* describes all the offers available in the MKBEEM-platform in terms of classes of services, e.g., service capabilities, non-functional attributes, etc. Service classes are generic in the sense that they are described independently from a specific provider. The source descriptions specify concrete services (i.e. provider offers) in terms of the service ontology. A further ontology is the linguistic domain ontology which assures an unambiguous interpretation of the user requests (see below in section 3).

The MKBEEM-mediation system allows to fill the gap between customer queries and diverse concrete providers offers. In a typical scenario, an end user submits to the MKBEEM-system a natural language query. The query is processed by a HUMAN LANGUAGE PROCESSING SERVER (HLP Server) which is in charge of *meaning extraction*: it analyzes the input string and converts the query into an *ontological formula* (OF) which is a language-independent formula containing the semantic information of the corresponding phrase in human language in terms of the service ontology. The OF is then sent to the DOMAIN ONTOLOGY SERVER (DOS). The DOS is responsible of storing, accessing and maintaining the ontologies used by the MKBEEM-system. It also provides the core reasoning mechanisms needed to support the mediation services. The DOS achieves a *contextual interpretation of the formula* using its knowledge about the application

domain. This task consists mainly in the identification of the offers (services) delivered by the MKBEEM-platform that "*best match*" the ontological formula. The aim here is to allow the users/applications to automatically discover the available services that best meet their needs, to examine their capabilities and to possibly complete missing information. The set of solutions computed by the DOS is sent back to the user to choose one solution and to complete the parameters, if any, that are missing. After this dialogue phase, the retained solution is sent back to the DOS to generate the *query plans*. A query plan contains information about the concrete services that are able to answer to the user query. Then, thanks to the technical information provided in the source descriptions, a query plan is translated into specific provider requests which are executed on the remote provider platforms.

Thus, the user poses queries in terms of the integrated knowledge (services and domain ontology) rather than directly querying specific provider information data-bases. This enables users to focus on *what* they want, rather than worrying about *how* and *from where* to obtain the answers.

Apart from the wrapping steps, which is no further considered in this paper (cf. [4] for more details), the MKBEEM-system relies on two mediation tasks, namely human language processing and service identification. These tasks are discussed in detail in the remainder of this paper.

3 Human Language Requests Analysis

Within MKBEEM, we currently cover three basic services of the tourism domain, i.e. train reservation, accommodation reservation, car rental as well as mail ordering of clothing. In all of these cases, human languages allow a wide range of expressions and the related linguistic ontology therefore contains all the necessary information. Another benefit of this is that it helps the user to specify as much parameters as needed in a single request, in natural language, thus avoiding tiresome form-filling. The combination of several requests (e.g. *I want to visit Paris and reserve a hotel next weekend*) is also possible.

To ensure that the generated, language neutral ontological formulas will contain all relevant information given by the user, the user request is treated in several interdependent steps [6].

Since the MKBEEM-prototype is multilingual, the first step is to identify the language of the user request. In the next step, it is analyzed and a language independent semantic graph is created. The linguistic analysis is based on a dependency syntax, a set of language dependent rules comparable to the Semantic Interpretation Rules of Discourse Representation Theory [7] and a set of language independent predicates. To ensure the ontological appropriateness of the generated semantic graph, it is checked by the linguistic domain ontology developed for this purpose.[4] Any inappropriate semantic graph is deleted from the set of possible solutions. Finally, in order to deal with travel dates etc. (especially in the tourism domain), temporal expressions which are relative to the

[4] This is done by PICSEL (ONTOCLASS) [8].

time of utterance (deictic elements like *now*, *today*, *in two hours*, *in five days*, *next Monday*, *at ten to eleven pm*) or incomplete or varying dates (*the 12th of April*, *on Good Friday* are transformed into the corresponding absolute temporal expression (if no exact time is specified, it is not generated):

temporal expression	transformation
now	*17.06.2003 13:56*
today	*17.06.2003*
in two hours	*17.06.2003 15:56*
in five days	*22.06.2003*
next Monday	*21.06.2003*
at ten to eleven pm	*17.06.2003 22:50*
the 12th of April	*12.04.2004*
on Good Friday	*9.04.2004*

The next step is the transformation of the internal semantic representation into the ontological formula, which is also understood by other modules. The concepts (and roles) differ considerably from the linguistic ontology due to the fact that linguistic expressions and semantic nuances are present in the semantic representation, which are not needed in the ontological formula. So for instance temporal/modal information (*I want to*/*I would like to*/*we will*/*we have to*) must be eliminated by the transformation. Further, different lexemes expressing a move (*go*/*arrive*/*depart*/*travel*/*be in*/*visit*) need to be mapped on the concept trip, which is the only move-concept of the service ontology (see below)

As an example we take a typical user request, like the following example 1:

Example 1. *"I'll arrive in Paris on Monday evening and I look for an accommodation with swimming pool."*

The request inquires information on public transport to Paris on (next) Monday evening (uttered on Tuesday, 17$^{\text{th}}$ June). After analyzing the sentence and processing the relative temporal information, we obtain an internal, language independent, semantic representation:[5]

Semantic representation 2. (simplified)

coord(coord1=x3005, coord2=x3006) &
arrival(destination=x3009, origin=u3010, situation=x3005, agent=x3013) &
speaker(theme=x3013) &
Paris(town=u3015, location=x3009) &
weekday~monday(date=x3005, wday=u3014) &
monthday~23(date=x3005, day=u3069) &
month~june(date=x3005, month=u3070) &
year~2003(date=x3005, year=u3071) &

[5] In this representation, we use predicates (in *italic*) and arguments (between parentheses) which indicate the semantic roles of the predicates. The predicates are linked via the variables of the arguments. For instance, the agent of the predicate arrival (x3013) is the speaker. Variables may link more than two predicates: x3005 links the situation of arrival with the date (Monday, 23$^{\text{rd}}$ June) and the time (18:00).

hour˜18(time=x3005, hour=u3072) &
minute˜0(time=x3005, minute=u3073) &
staying(agent=x3021, situation=x3006, place=x3022, means=x3023,
leisure=x3024) &
speaker(theme=x3021) &
accomodationorg(city=x3022, theme=x3023, leisure=x3024) &
swimmingPool(type=x3024).

As users are not directly concerned by the organization of data provided by information systems (in our case train, car rental, tourism), the main difficulty is to map efficiently the *user concepts* (*go*, *arrive*, *depart*, *take a train*, etc.), identified by the HLP, onto the information system (IS) concepts. Since some user requests are complex utterances, mixing motion verbs with absolute or relative time and space representation, the linguistic ontology is first used to constrain the parser during the construction of the linguistic formulas and to reduce the ambiguity ([9], cf. also [10]). In a next step irrelevant information (from an application point of view) must be pruned to produce a new formula compliant to the DOS (cf. section 4), devoted identify the service and to plan the data-base queries.

The linguistic ontology has been designed using the experience and knowledge gained in a previous project using description logics (PICSEL[6]), and which tools have been enriched to fit the needs of linguistic analyzer.

Usually, ontologies are organized as directed graphs and use multiple inheritance. In consequence the more general concepts subsume the more specific. In contrast to superordinates which are less specific concepts, the greatest common subsumee (GCS) are more complete. Our experience, however, shows, that IS concepts are rather GCS than superordinates.

As outlined in [9] we use a common formalism for information representation (ontology). The ontologies are represented in PICSEL, where concepts are unary predicates and roles binary predicates joining two concepts or a concept and a constant. The common inter-module communication language is CARIN-$\mathcal{ALN}$[7] which is in the framework of Description Logics. As a consequence the HLP must transform utterances into formulas (using the inter-module communication language)

PICSEL ontologies are organized as directed graphs and use multiple inheritance. Thus in PICSEL (and other DLs) the more generic concepts subsume the more specific ones. In natural languages, however, more general concepts

[6] "PICSEL is an information integration system over sources that are distributed and possibly heterogeneous. The approach which has been chosen in PICSEL is to define an information server as a knowledge-based mediator in which CARIN is used as the core logical formalism to represent both the domain of application and the contents of information sources relevant to that domain." [11]:383, [8].

[7] CARIN is a family of theoretical languages for knowledge representation, CARIN-$\mathcal{ALN}$ is the most expressive description logic for which subsumption and satisfiability are polynomial [11].

combine features of more specific ones. In consequence, the greatest common subsumees (GCS) are the best candidates to represent these more general concepts. Our experience shows that information system applications should rather use GCS than concepts of the linguistic sub-ontology (LSO) in order to keep the power of inheritance and to manage a more generic notion at the same time.

Discrepancies between the semantic representation (of the user request) and the main ontology must thus be bridged: The semantic representations (graphs) are using the LSO (i.e. concepts and roles defined in the LSO). To obtain the ontological formula, we need to rewrite this representation in service ontology (SO) terms. In order to achieve this, the principal rewriting rule is to replace the LSO concept (as found during the syntactical-semantic analysis) by the GCS concept of the SO.

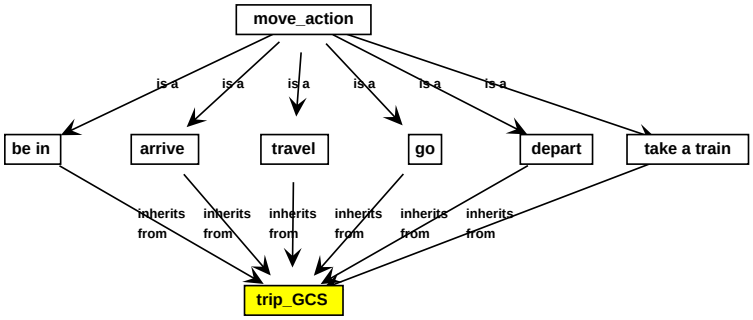


Fig. 2. Links between linguistic ontology and service ontology

As figure 2 shows, the service ontology concept trip_GCS[8] inherits all existing information from the (linguistic) concepts go, arrive etc., which express the meanings of the verbs in question. The motion verb therefore can be rewritten using the GCS (in this case trip_GCS). The resulting formula can now be interpreted correctly within the service ontology. Taking our example 1 (page 5), the semantic representation 2 (page 6) is thus transformed into the corresponding ontological formula 3 in service ontology terms. (cf. figure 3).[9]

[8] The suffix _CGS is used only for clarity.

[9] An ontological formula is a particular kinds of conjunctive queries expressed on unary and binary predicates. Roughly speaking, the ontological formula 3 of our example defines two concepts:

- trips, specified through the variable V5609, whose destination is *Paris* (denoted by properName_Paris) and whose arrival date is: *Monday, 23th of June 2003* at *18:00*.
- accommodations, specified through the variable V5610, offering a *swimming pool* among their leisure facilities.

Hence, this ontological formula 3 expresses the fact the user is interested by the instances of these two concepts.

Ontological formula 3.

(trip)(V5609),
  (arrPlace)(V5609, properName_Paris),
(date)(C63),
  (weekday)(C63, monday),
  (day)(C63, 23),
  (month)(C63, june),
  (year)(C63, 2003),
  (arrDate)(V5609, C63),
(time)(C64),
  (hour)(C64, 18),
  (minute)(C64, 0),
  (arrTime)(V5609, C64),
(accommodation)(V5610),
  (leisure)(V5610, swimmingPool)

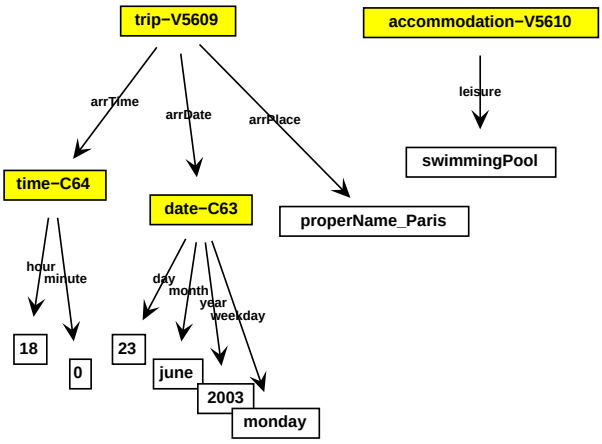


Fig. 3. Visualization of the ontological formula used for the service identification

4 Service Identification

In MKBEEM, service identification is achieved by means of a *dynamic service discovery* reasoning mechanism. Dynamic service discovery is used in association with the PICSEL system to achieve the reasoning tasks in the DOS. The complementary roles of these two complex logical reasoning constitutes the description logic core for query processing in the MKBEEM-system. They are in fact two different instances of the problem of rewriting concepts using terminologies [12].

The following example illustrates the interest of the *service discovery* reasoning mechanism.

Let us consider an e-commerce platform that delivers the following four offers:

- hotel, which allows to consult a list of hotels.
- apartment, which allows to consult a list of apartments.
- timetable1, which allows to consult a journey given the departure place, the arrival place, the departure date and the departure time.
- timetable2, which allows to consult a journey given the departure place, the arrival place, the arrival date and the arrival time.

Let us assume that, according to architecture of the MKBEEM-ontology, these offers are formally described in a given service ontology. Consider now, the example 1 (above page 5) and the ontological formula 3 (page 8) created by HLP Server. Now the *service discovery* is used by the DOS to identify the corresponding relevant service(s) in the service ontology. This task is achieved in two steps:

1. Converting an ontological formula F into a concept description Q_F:
   This task depends on the structure of the ontological formula and on the expressive power of the target language. In the context of the MKBEEM-project, the current ontological formulas generated by the HLP SERVER have relatively simple structures that can be described using the small description logic $\mathcal{FL}_0 \cup \{(\geq nR)\}$. This logic contains the concept conjunction constructor ($\sqcap$), the universal role quantification constructor ($\forall R.C$) and the minimal number restriction constructor ($\geq nR$). In this case, we can achieve this task by computing the so-called *most specific concept* [13] corresponding to the ontological formula.
   The concept description Q_{OF1} corresponding to the ontological formula *OF1* given in the previous example is:

$$
\begin{aligned}
Q_{OF1} \doteq\ & \mathsf{accommodation} \\
& \sqcap (\geq 1\ \mathsf{leisure}) \\
& \sqcap (\forall\ \mathsf{leisure\ string}) \\
& \sqcap \mathsf{trip} \\
& \sqcap (\geq 1\ \mathsf{arrPlace}) \\
& \sqcap (\forall\ \mathsf{arrPlace\ string}) \\
& \sqcap (\geq 1\ \mathsf{arrDate}) \\
& \sqcap (\forall\ \mathsf{arrDate}\ (\mathsf{date} \sqcap (\geq 1\ \mathsf{day}) \sqcap (\forall\ \mathsf{day\ integer}) \\
& \qquad \sqcap (\geq 1\ \mathsf{year}) \sqcap (\forall\ \mathsf{year\ integer}) \\
& \qquad \sqcap (\geq 1\ \mathsf{month}) \sqcap (\forall\ \mathsf{month\ string}) \\
& \qquad \sqcap (\geq 1\ \mathsf{weekday}) \sqcap (\forall\ \mathsf{weekday\ string}))) \\
& \sqcap (\geq 1\ \mathsf{arrTime}) \\
& \sqcap (\forall\ \mathsf{arrTime}\ (\mathsf{time} \sqcap (\geq 1\ \mathsf{hour}) \sqcap (\forall\ \mathsf{hour\ integer}) \\
& \qquad \sqcap (\geq 1\ \mathsf{minute}) \sqcap (\forall\ \mathsf{minute\ integer})))
\end{aligned}
$$

2. Selecting the relevant services:

This problem can be stated as follows: given a user query Q_F and an ontology of services T, find a description E, built using (some) of the names defined in T, such that E contains as much as possible of common information with Q_F and as less as possible of extra information with respect to Q_F. We call such a rewriting E a *best cover* of Q_F using T. Therefore, our goal is to rewrite a description Q_F into the closest description expressed as a conjunction of (some) concept names in T.

A best cover E of a concept Q using T is defined as being any conjunction of concept names occurring in T which shares some common information with Q, is consistent with Q and minimizes, in this order, the extra information in Q and not in E and the extra information in E and not in Q. Once the notion of a best cover has been formally defined, the second issue to be addressed is how to find a set of services that best covers a given query. This problem, called *best covering problem*, can be stated as follows: given an ontology T and a query description Q, find all the best covers of Q using T.

More technical details about the best covering problem can be found in [14,15]. To sum up, the main results that have been reached are:

- The precise formalisation of the best covering problem in the framework of languages where the difference operation is semantically unique (e.g., the description logic $\mathcal{FL}_0 \cup \{(\geq nR)\}$).
- A study of complexity showed that this problem is NP-Hard ([16]).
- A reduction of the best covering problem to the problem of computing the minimal transversals with minimum cost of a weighted hypergraph.
- Based on hypergraph theory, a sound and complete algorithm that solves the best covering problem was designed and implemented.

Continuing with the example, we obtain the following result from the DOS:

	identified services	**rest**	**missing information**
Solution 1	timetable2,	leisure	depPlace
	apartment	—	numberOfRooms, apartmentCategory
Solution 2	timetable2,	leisure	depPlace
	hotel	—	numberOfBeds, hotelCategory

Table 1. Results from the DOS

These solutions correspond to the combinations of services that best match the ontological formula *OF1*. For each solution, the DOS computes the extra information (column *missing information*) brought by the services but not contained in the user query. The column *rest* contains the extra information (leisure) contained in the user query and not provided by any services. This means that, in the proposed solutions the requirement concerning the leisure is not taken into account.

To continue with the example, assume that the user chooses the first solution (timetable2, apartment). Then, he is asked to complete the missing information: the departure place, the apartment category and the number of rooms the user wants in the apartment. The result is a global query Q, expressed as a service formula, that will be sent to the Picsel system to identify the providers which are able to answer to this query.

5 Conclusion

In this paper we have described the successful implementation of multilingual mediation system, based on knowledge which is coded in ontologies. It shows, how after the identification of the language, a user request is analysed and transformed into an language independent ontological representation. This representation is used to identify the service/product the user wants to consult/buy by the help of service ontologies. Existing parameters are extracted, missing ones request in a subsequent step. Finally the data base of the appropriate content provider is contacted to present the user the results of his initial requests.

Acknowledgements

The authors would like to thank all their colleagues in the MKBEEM consortium for the excellent cooperation as well as the European Commission for supporting the reported work. We are grateful to three anonymous reviewers for valuable hints for improvement of an earlier version of this paper. All remaining errors are our own.

References

1. Fensel, D.: *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, Heidelberg (2001)
2. Léger, A., Michel, G., Barrett, P., Gitton, S., Gómez-Pérez, A., Lehtola, A., Mokkila, K., Rodrigez, S., Sallantin, J., Varvarigou, T., Vinesse, J.: Ontology domain modeling support for multi-lingual services in E-Commerce: MKBEEM. In: *Proceedings of European Conference on Artificial Intelligence (ECAI 2000), Berlin. Workshop on Applications of Ontologies and Problem-Solving Methods*. (2000)
3. Gómez-Pérez, A.: Requirement, Choice of a Knowledge Representation and Tools. Technical report, Universidad Politécnica de Madrid, Madrid (2001)
4. MKBEEM: The MKBEEM-project. http://www.mkbeem.com/ (2000-2002)
5. Corcho, O., Gómez-Pérez, A., Léger, A., Rey, C., Toumani, F.: An Ontology-based Mediation Architecture for e-commerce applications. In: *Intelligent Information Systems 2003 (IIS2003), Zakopane, Poland*. Advances in Soft Computing, Heidelberg, Springer (2003)
6. Lehtola, A., Heinecke, J., Bounsaythip, C.: Intelligent Human Language Query Processing in MKBEEM. In: *Proceedings of the Workshop on Ontologies and Multilinguality in User Interface. HCII 2003, Creta, Greece*. (forth)

7. Kamp, H., Reyle, U.: *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Studies in Linguistics and Philosophy 42. Kluwer, Dordrecht (1993)
8. PICSEL: The PICSEL-project. http://www.lri.fr/~picsel/ (1999-2001)
9. Heinecke, J., Cozannet, A.: Ontology-Driven Information Retrieval. a proposal for multilingual user requests (2003) Paper given at the Workshop on Ontological Knowledge and Linguistic Coding at the 25th annual meeting of the German Linguistics, Feb. 26-28. 2003.
10. Guarino, N.: Towards a common ontology for multilingual information access and integration. Paper held at the Third Meeting of SIG on Intelligent Information Agents. Barcelona (1999)
11. Goasdoué, F., Lattes, V., Rousset, M.C.: The Use of CARIN Language and Algorithms for Information Integration. The PICSEL-Project. *International Journal of Cooperative Information Systems* **9:4** (2000)
12. Baader, F., Küsters, R., Molitor, R.: Rewriting Concepts Using Terminologies. In: *Proceedings of the Seventh International Conference on Knowledge Representation and Reasoning, Colorado, USA.* (2000) 297–308
13. Donini, F., M. Lenzerini, D. Nardi, A.S.: Reasoning in description logics. In Brewka, G., ed.: *Foundation of Knowledge Representation*, CSLI-Publications (1996) 191–236
14. Hacid, M., Leger, A., Rey, C., Toumani, F.: Computing concept covers: A preliminary report. In: *International Workshop on Description Logics (DL 2002). Toulouse, France.* (2002)
15. Hacid, M., Leger, A., Rey, C., Toumani, F.: Dynamic discovery of e-services: A description logics based approach. In: *Proceedings of the 18th French conference on advanced databases (BDA), Paris.* (2002) 21–25
16. Hacid, M.S., Léger, A., Rey, C., Toumani, F.: Dynamic Discovery of E-services. A Description Logics Based Approach. Technical report, LIMOS, Clermont-Ferrand (2002)

Axiomatizing WordNet Glosses in the OntoWordNet Project

Aldo Gangemi[1], Roberto Navigli[2], Paola Velardi[2]

[1]Laboratory for Applied Ontology, ISTC-CNR,
viale Marx 15, 00137 Roma, Italy
gangemi@ip.rm.cnr.it

[2] Dipartimento di Informatica, University of Roma "La Sapienza"
via Salaria 113, 00198 Roma, Italy
{navigli,velardi}@dsi.uniroma1.it

Abstract. In this paper we present a progress report of the OntoWordNet project, a research program aimed at achieving a formal specification of WordNet. Within this program, we developed a hybrid bottom-up top-down methodology to automatically extract association relations from WordNet, and to interpret those associations in terms of a set of conceptual relations, formally defined in the DOLCE foundational ontology. Preliminary results provide us with the conviction that a research program aiming to obtain a consistent, modularized, and axiomatized ontology from WordNet can be completed in acceptable time with the support of semi-automatic techniques.

1. Introduction

The number of applications where WordNet (WN) is being used as an ontology rather than as a mere lexical resource seems to be ever growing. Indeed, WordNet contains a good coverage of both the lexical and conceptual palettes of the English language. However, WordNet is serviceable as an ontology (in the sense of a *theory* expressed in some *logical language*) if some of its lexical links are interpreted according to a formal semantics that tells us something about the way we use a lexical item in some context for some purpose. In other words, we need a *formal specification of the conceptualizations that are expressed by means of WordNet's synsets*[1]. A formal specification requires a clear semantics for the primitives used to export WordNet

[1] Concept names in WordNet are called *synsets*, since the naming policy for a concept is a set of synonym words, e.g. for sense 1 of car: { car, auto, automobile, machine, motorcar }. In what follows, WN concepts are also referred to as synsets.

information into an ontology, and a methodology that explains how WordNet information can be bootstrapped, mapped, refined, and modularized during the export procedure.

The formal specification of WordNet is the objective of the so-called OntoWordNet research program, started two years ago at the ISTC-CNR, and now being extended with other partners, since collaborations have been established with the universities of Princeton, Berlin and Roma. The program is detailed in section 2, where we outline the main objectives and current achievements.

In this paper we describe a joint ongoing work of ISTC-CNR and the University of Roma that has produced a methodology and some preliminary results for adding *axioms* (DAML+OIL "restrictions") to the concepts derived from WordNet synsets. The methodology is hybrid because it employs both top-down techniques and tools from formal ontology, and bottom-up techniques from computational linguistics and machine learning. Section 3 presents a detailed description of the methodology.

The preliminary results, presented in section 4, seem very encouraging, and provide us with the conviction that a research program aiming to obtain a consistent, modularized, and axiomatized ontology from WordNet can be completed in acceptable time with the support of semi-automatic techniques.

2. The OntoWordNet research program: objectives, assumptions, and first achievements

The OntoWordNet project aims at producing a formal specification of WordNet as an axiomatic theory (an *ontology*). To this end, WordNet is reorganized and enriched in order to adhere to the following commitments:

- *Logical commitment*. WordNet synsets are transformed into logical types, with a formal semantics for lexical relations. The WordNet lexicon is also separated from the logical namespace.
- *Ontological commitment*. WordNet is transformed into a general-purpose ontology library, with explicit categorial criteria, based on formal ontological distinctions (Gangemi et al. 2001). For example, the distinctions enable a clear separation between (kinds of) concept-synsets, relation-synsets, meta-property-synsets, and enable the instantiation of individual-synsets. Moreover, such formal ontological principles facilitate the axiomatic enrichment of the ontology library.
- *Contextual commitment*. WordNet is modularized according to knowledge-oriented domains of interest. The modules constitute a partial order.
- *Semiotic commitment*. WordNet lexicon is linked to text-oriented (or speech act-oriented) domains of interest, with lexical items ordered by preference, frequency, combinatorial relevance, etc.

A set of *logical commitments* has been introduced in WordNet through methodological assumptions that are described in (Gangemi et al. 2002). The hyperonymy relation in WN is basically interpreted as *formal subsumption*, although hyperonymy for concepts referring to individuals (geographical names, characters,

some techniques, etc.) is interpreted as *instantiation.* This will be referred as *assumption A1* ("hyperonymy as synset subsumption"). For example, the concept *retrospective#1* has the hyperonym *art_exhibition#1*, which is logically represented as:

$\forall x.\ \text{Retrospective}(x) \rightarrow \text{Art\_Exhibition}(x)$,

while the hyperonymy link between the *gemini#1* and *constellation#1* is represented as an *instantiation*:

Constellation(*Gemini*)

WordNet's *ontological commitments* are more demanding to be explicitated, but many results are already available. For example, an incremental methodology has been adopted, reusing the DOLCE foundational ontology (Gangemi et al. 2002), in order to revise or to reorganize WordNet synset taxonomies and relations (see also paragraph 3.2.1). Substantial work has been done on the refinement of the hyponym/hyperonym relations, which have being investigated since several years. WordNet synonymy is a relation between words, not concepts, therefore we should assume that the synonymy relation (*synsets* in WordNet) is an *equivalence class* of words (or phrases), sharing the same *meaning* within an ontology. Consequently, two words are synonyms when their intended meaning in WordNet is the same. This will be referred to as *assumption A2* ("synset as meaning equivalence class").
However, we have no formal definition of words in WordNet that allows us to create equivalence classes (synsets) analytically (i.e., to state *semantic equivalences*). Instead, we have pre-formal synsets that have been validated by lexicographers with an intuition that *could* be formalized as semantic equivalence. Part of this intuition is conveyed by textual definitions (called *glosses*). No claim of completeness is made though. This will be referred as *assumption A3* ("glosses as axiomatizations"). In this paper we are trying to formalize such intuition.
A related assumption that we make is that words in glosses are used in a way consistent to the WordNet synsets. This will be referred as *assumption A4* ("glosses are synset-consistent"). A4 lets us assume also that the informal theory underlying synsets, hyperonymy relations, and glosses, can be formalized against a finite signature (the set of WN synsets), and a set of axioms derived from the associations (*A-links*) between any synset *S* and the synsets that can be associated to the words used in the gloss of *S*. This is dependent on A3 and A4, and may be referred as *assumption A5* ("*A-links* as conceptual relations").
The revision of WordNet synset taxonomies is still ongoing, but it is already usable to carry out novel experiments. For example, the WEBKB-2[2] project is using the preliminary results of our work.

Contextual and semiotic commitments are very partially implemented, although some resources and the methodologies to exploit them are available. For example, contextual information could be extracted using the so-called *domain labels* defined

[2] http://meganesia.int.gu.edu.au/~phmartin/WebKB/doc/wn

in (Miller et al. 1993) and (Magnini and Cavaglia, 2000). Domain labels have been associated to WordNet 1.6 synset, and we are currently analyzing and refining this information.
Domain labels are being exploited in order to create a partial order of ontological modules that is consistent with the actual use of the lexicon within real world corpora. To this purpose, we are using both foundational ontologies (top-down reorganization), and Web catalogues (bottom-up reorganization).

Figure 1 shows the "layers" in which the OntoWordNet ontology library is being organized. The foundational layer contains modules including domain-independent concepts, relations, and meta-properties. The core layer contains modules including generic concept and relations for a given domain of interest. The domain layer contains modules including domain-oriented instances, concepts, and relations. This layer can be automatically populated by an ontology extension technique, implemented in the OntoLearn system (Navigli et al. 2003).

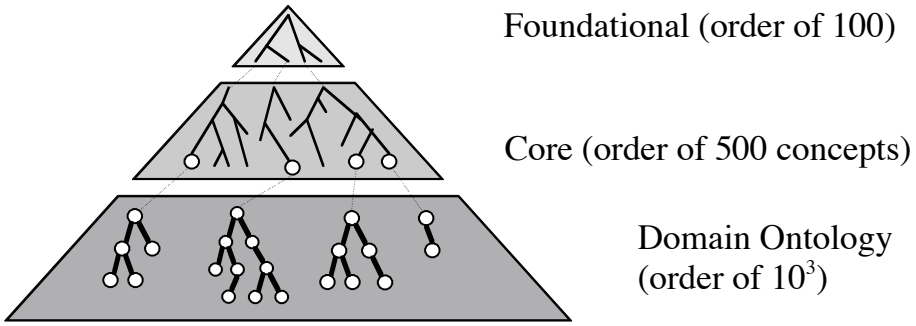


Figure 1. The three levels of generality of a Domain Ontology.

3. Semi-automatic axiomatization of WordNet

The task of axiomatizing WordNet, starting from assumptions A1-A5 outlined in the previous section, requires that the informal definition in a synset gloss be transformed in a logical form. To this end, first, words in a gloss must be disambiguated, i.e. replaced by their appropriate synsets. This first step provides us with pairs of generic semantic associations (A-links) between a synset and the synsets of its gloss. Secondly, A-links must be interpreted in terms of more precise, formally defined semantic relations. The inventory of semantic relations is selected or specialized from the foundational ontology DOLCE, as detailed later, since in WordNet only a limited set of relations are used, that are partly ontological, partly lexical in nature. For example, part_of (*meronymy* in WordNet) and kind_of (*hyponymy* in WordNet) are typical semantic relations, while *antonymy* (e.g. *liberal* and *conservative*) and *pertonymy* (e.g *slow* and *slowly*) are lexical relations. Furthermore, WordNet relations are not axiomatized, nor are they used in a fully consistent way.
To summarize, the objective of the method described in this section is to:

- automatically extract a number of semantic relations implicitly encoded in WordNet, i.e. the relations holding between a synset and the synsets in its gloss.
- (semi)-automatically interpret and axiomatize these relations.

For example, sense 1 of *driver* has the following gloss "the operator of a motor vehicle". The appropriate sense of *operator* is #2: *operator, manipulator* ("an agent that operates some apparatus or machine"), while motor vehicle is monosemous: *motor vehicle, automotive vehicle* ("a self-propelled wheeled vehicle that does not run on rails").
After automatic sense disambiguation, we (hopefully) learn that there exists an A-link between driver#1 and operator#2, and between *driver#1* and *motor vehicle#1*. Subsequently, given a set of axiomatized semantic relations in DOLCE, we must select the relation that best fits the semantic restrictions on the relation universes (domain and co-domain, or range). For example, given an A-link between *driver#1* and *motor vehicle#1*, the best fitting relation is *agentive-co-participation* (Figure 2), whose definition is:

$$AG\_CO\_PCP(x,y) =_{df} CO\_PCP(x,y) \sqcap Agentive\_Physical\_Object(x) \sqcap \sqcap Non\_Agentive\_Functional\_Object(y)$$

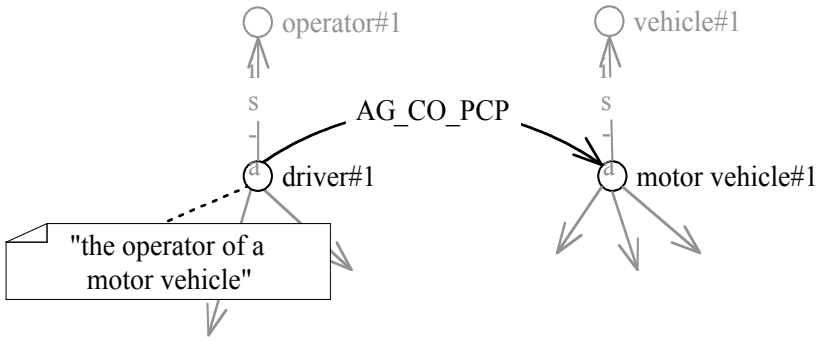


Figure 2. An example of semantic relation.

The definition says that *agentive co-participation* is a relation of mutual participation (participation of two objects in the same event), with the domain restricted to "Agentive_Physical_Object" and the range restricted to "Non_Agentive_Functional_Object".
Domain and range in a conceptual relation definition are established in terms of the DOLCE ontology. Consequently, another necessary step of our method is to re-link at least some of the higher level nodes in WordNet with the DOLCE upper ontology.
In the following sub-sections we detail the procedures for gloss disambiguation, WordNet re-linking, and selection of conceptual relations.

3.1 Bottom-up learning of association links.

The first step is a bottom-up procedure that analyses the NL definitions (glosses) in WordNet and creates the A-links.
For each gloss (i.e., linguistic concept definition), we perform the following automatic tasks:
a) POS-tagging of glosses (using the ARIOSTO NL processor) and extraction of *relevant* words;
b) Disambiguation of glosses by the algorithm described hereafter;
c) Creation of explicit "association" links (A-links) from synsets found in glosses to synsets to which glosses belong.

3.1.1 Description of the gloss disambiguation algorithm

We developed a greedy algorithm for gloss disambiguation that relies on a set of heuristic rules and is based on multiple, incremental iterations. A simplified formal description of the algorithm is in Figure 3.

The algorithm takes as input the synset *S* whose gloss *G* we want to disambiguate.

Two sets are used, *P* and *D*. *D* is a set of disambiguated synsets, initially including only the synset S. *P* is a set of terms to be disambiguated, initially containing all the terms from gloss *G* and from the glosses {G'} of the direct hyperonyms of *S*. As clarified later, adding {G'} provides a richer context for semantic disambiguation. The term list is obtained using our NL processor to lemmatize words, and then removing irrelevant words. We use standard information retrieval techniques (e.g stop words) to identify irrelevant terms.

When, at each iteration of the algorithm, we disambiguate some of the terms in *P*, we remove them from *P* and add their interpretation (i.e. synsets) to the set *D*. Thus, at each step, we can distinguish between *pending* and *disambiguated* terms (respectively the sets *P* and *D*). Notice again that *P* is a set of terms, while *D* contains synsets.

a) Find monosemous terms

The first step of the algorithm is to remove monosemous terms from *P* (those with a unique synset) and include their unique interpretation in the set *D*.

b) Disambiguate polysemous terms

Then, the core iterative section of the algorithm starts. The objective is to detect *semantic relations* between some of the synsets in *D* and some of the synsets associated to the terms in *P*. Let *S'* be a synset in *D* (an already chosen interpretation of term *t'*) and *S"* one of the synsets of a polysemous term *t"* ∈ *P* (i.e., *t"* is still ambiguous). If a semantic relation is found between *S'* and *S"*, then *S"* is added to *D* and *t"* is removed from *P*.

To detect semantic relations between *S'* and *S"*, we apply a set of heuristics grouped in two classes, *Path* and *Context*, described in what follows. Some of these heuristics have been suggested in (Milhalcea, 2001),

Path heuristics

The heuristics in class Path seek for *semantic patterns* between the node *S'* and the node *S"* in the WordNet semantic network. A *pattern* is a chain of nodes (synsets) and arcs (directed semantic relations), where *S'* and *S"* are at the extremes.

Formally, we define $S' \xrightarrow{R}{}^{n} S''$ as $S' \xrightarrow{R} S_1 \xrightarrow{R} \ldots \xrightarrow{R} S_n \equiv S''$, that is a chain of n instances of the relation R. We also define $\xrightarrow{R_1,R_2}$ as $\xrightarrow{R_1}\ \xrightarrow{R_2}$.

The symbols: $\xrightarrow{@}, \xrightarrow{\sim}, \xrightarrow{\#}, \xrightarrow{\%}, \xrightarrow{\&}$ respectively represent the following semantic relations coded in WordNet 1.6: *hyperonimy* (kind_of), *hyponymy* (has kind) *meronymy* (part_of), *holonymy* (has_part), and *similarity*. Similarity is a generic relation including near synonyms, adjectival clusters and antonyms. Finally, the *gloss* relation $S \xrightarrow{gloss} T$ indicates that the gloss of S includes a term t, and T is one of the synsets of *t*.

We use the following heuristics to identify semantic paths ($S' \in D$, $S'' \in Synsets(t'')$, $t'' \in P$):

1 *Hyperonymy path*: if $S' \xrightarrow{@}{}^{n} S''$ choose *S"* as the right sense of *t"* (e.g., *canoe#1* $\xrightarrow{@}{}^{2}$ *boat#1*, i.e. a *canoe* is a kind of *boat*);

2 *Hyperonymy/Meronymy path*: if $S' \xrightarrow{@,\#}{}^{n} S''$ choose *S"* as the right sense of *t"* (e.g., *archipelago#1* $\xrightarrow{\#}$ *island#1*);

3 *Hyponymy/Holonymy path*: if $S' \xrightarrow{\sim,\%}{}^{n} S''$ choose *S"* as the right sense of *t"* (e.g., *window#7* $\xrightarrow{\%}$ *computer screen#1*);

4 *Adjectival Similarity*: if *S"* is in the same adjectival cluster than *S'*, choose *S"* as the right sense of *t"*.

5 *Parallelism*: if exists a synset *T* such that $S' \xrightarrow{@} T \xrightarrow{@} S''$, choose *S"* as the right sense of *t"* (for example, *background#1* $\xrightarrow{@}$ *scene#3* $\xrightarrow{@}$ *foreground#2*);

Context heuristics

The Context heuristics use several available resources to detect co-occurrence patterns in sentences and contextual clues to determine a semantic proximity between *S'* and *S"*. The following heuristics are defined:

1 *Semantic co-occurrences*: word pairs may help in the disambiguation task if they always co-occur with the same senses within a tagged corpus. We use three resources in order to look for co-occurrences, namely:

- the *SemCor corpus*, a corpus where each word in a sentence is assigned a sense selected from the WordNet sense inventory for that word; an excerpt of a SemCor document follows:

  Color#1 was delayed#1 until 1935, the widescreen#1 until the early#1 fifties#1.
  Movement#7 itself was#7 the chief#1 and often#1 the only# attraction#4 of the primitive#1 movies#1 of the nineties#1.

- the *LDC corpus*, a corpus where each document is a collection of sentences having a certain word in common. The corpus provides a sense tag for each occurrence of the word within the document. Unfortunately, the number of documents (and therefore the number of different tagged words) is limited to about 200. An example taken from the document focused on the noun *house* follows:

  *Ten years ago, he had come to the **house#2** to be interviewed.*
  *Halfway across the **house#1**, he could have smelled her morning perfume.*

- *gloss examples:* in WordNet, besides glosses, examples are sometimes provided containing synsets rather than words. From these examples, as for the LDC Corpus, a co-occurrence information can be extracted. With respect to the LDC corpus, WordNet provides examples for thousands of synsets, but just a few for the same word. Some examples follow:

  *"Overnight **accommodations#4** are available."*
  *"Is there **intelligent#1** life in the universe?"*
  *"An **intelligent#1** question."*

As we said above, only the SemCor corpus provides a sense for each word in a pair of adjacent words occurring in the corpus, while LDC and gloss examples provide the right sense only for one of the terms.
In either case, we can use this information to choose the synset *S"* as the interpretation of *t"* if the pair *t' t"* occurs in the gloss and there is an agreement among (at least two of) the three resources about the disambiguation of the pair *t' t"*. For example:
*[…] Multnomah County may be short of general assistance money in its budget to handle an unusually high **summer#1 month#1**'s need […].*
*Later#1, Eckenfelder increased#2 the efficiency#1 of treatment#1 to between 75 and 85 percent#1 in the **summer#1 months#1**.*
are sentences respectively from the LDC Corpus and SemCor. Since there is a full agreement between the resources, one can easily disambiguate *summer* and *months* in the gloss of *summer_camp#1*: "*a site where care and activities are provided for children during the **summer months***".

2 *Common domain labels*: Domain labels are the result of a semiautomatic methodology described in (Magnini and Cavaglia, 2000) for assigning domain labels (e.g. *tourism, zoology, sport..*) to WordNet synsets[3]. This information can be exploited to disambiguate those terms with the same domain labels of the start synset *S*. Notice that a synset can be marked with many domain labels, therefore the algorithm selects the interpretation *S"* of *t"* if the following conditions hold together (the *factotum* label is excluded because it is a sort of topmost domain):

- *DomainLabels*(*S"*) \ { factotum } ∩ *DomainLabels* (*S*) \ { factotum };
- There is no other interpretation *S"'* of *t"* such that *DomainLabels* (*S"'*) \ { factotum } ∩ *DomainLabels* (*S*) \ { factotum }.

For example, *boat#1* is defined as "*a small vessel for travel on water*", both *boat#1* and *travel#1* belong to the *tourism* domain and no other sense of *travel* satisfies the conditions, so the first sense of *travel* can be chosen; similarly, *cable car#1* is defined as "*a conveyance for passengers or freight on a cable railway*", both *cable car#1* and *conveyance#1* belong to the *transport* domain and no other sense of *conveyance* satisfies the conditions, so the first sense of *conveyance* is selected.

c) Update *D* and *P*

During each iteration, the algorithm applies all the available heuristics in the attempt of disambiguating some of the terms in *P*, using all the available synsets in *D*. While this is not explicit in the simplified specification of Figure 3, the heuristics are applied in a fixed order reflecting their importance, that has been experimentally determined. For example, Context heuristics are applied after Path heuristics 1-5. At the end of each iterative step, new synsets are added to *D*, and the correspondent terms are deleted from *P*. The next iteration makes use of these new synsets in order to possibly disambiguate other terms in *P*. Eventually, either *P* becomes empty, or no new semantic relations can be found.

When the algorithm terminates, *D* \ { *S* } can be considered a first approximation of a *semantic definition of S*. For mere gloss disambiguation purposes, the tagged terms in the hyperonyms' gloss are discarded, so that the resulting set (*GlossSynsets*) now contains only interpretations of terms extracted from the gloss of *S*. At this stage, we can only say that there is a semantic relation (A-link) between *S* and each of the synsets in *GlossSynsets* .

A second, more precise approximation of a sound ontological definition for *S* is obtained by determining the nature of the A-links connecting *S* with each concept in *D* \ { *S* }. This is an ongoing task and is discussed in Section 4.

[3] Domain labels have been kindly made available by the IRST to our institution for research purposes.

3.1.2 A running example

In the following, we present a sample execution of the algorithm on sense 1 of *retrospective*. Its gloss defines the concept as "*an exhibition of a representative selection of an artist's life work*", while its hyperonym, *art exhibition#1*, is defined as "*an exhibition of art objects (paintings or statues)*". Initially we have:

D = { *retrospective#1* }
P = { *work, object, exhibition, life, statue, artist, selection, representative, painting, art* }

The application of the monosemy step gives the following result:

D = { *retrospective#1, statue#1, artist#1* }
P = { *work, object, exhibition, life, selection, representative, painting, art* }

because *statue* and *artist* are monosemous terms in WordNet. During the first iteration, the algorithm finds three matching paths:

retrospective#1 $\square^{@}_{2}$ *exhibition#2*, *statue#1* $\square^{@}_{3}$ *art#1* and *statue#1* $\square^{@}_{6}$ *object#1*

this leads to:

D = { *retrospective#1, statue#1, artist#1, exhibition#2, object#1, art#1* }
P = { *work, life, selection, representative, painting* }

During the second iteration, an hyponymy/holonymy path is found:

art#1 $\square^{\sim}_{2}$ *painting#1* (painting is a kind of art)

D = { *retrospective#1, statue#1, artist#1, exhibition#2, object#1, art#1, painting#1* }
P = { *work, life, selection, representative* }

Since no new paths are found, the third iteration makes use of the LDC Corpus to find the co-occurrence "*artist life*", with sense 12 of *life* (*biography*, *life history*):

D = { *retrospective#1, statue#1, artist#1, exhibition#2, object#1, art#1, painting#1, life#12* }
P = { *work, selection, representative* }

Notice that, during an iteration, the context heuristics are used only if the path heuristics fail.

The algorithm stops because there are no additional matches. The chosen senses concerning terms contained in the hyperonym's gloss were of help during disambiguation, but are now discarded. Thus we have:

GlossSynsets(retrospective#1) = { *artist#1, exhibition#2, life#12* }

```
DisambiguateGloss(S)

{G already disambiguated? }
if (GlossSynset(S) ≠ Ø) return

{ S is the starting point }
D := { S }
{ disambiguation is applied the terms within the gloss of S
  and the glosses of its direct hyperonyms }
P := Gloss(S) ∪ Gloss(Hyper(S))

{look for synsets associated to monosemous terms in P }
M := SynsetsFromMonosemousTerms(P)
D := D ∪ M
{ 'Terms' returns the terms contained in the gloss of M }
P := P \ Terms(M)

LastIteration:=D
{ until there is some heuristic to apply }
while(LastIteration ≠ Ø)
   NS := Ø { new chosen synsets for disambiguating
            terms in the gloss of S }

      { for each just disambiguated synset S'}
      foreach (S' ∈ LastIteration)
         { look for connections between S'
           and the synsets to disambiguate }
         NS := NS ∪ Path-heuristics(S', P)

         NS := NS ∪ Context-heuristics(S', P)

      { D now contains all the new chosen synsets from
        the last iteration }
      D := D ∪ NS
      { remove the terms contained in the gloss of NS }
      P := P \ Terms(NS)
      { these results will be used
        in the next iteration }
      LastIteration := NS

{ stores the synsets chosen for some terms
  in the gloss of S }
foreach S' ∈ D
   if (Terms(S') ∩ Gloss(S) ≠ Ø)
      GlossSynsets(S) := GlossSynsets(S) ∪ { S' }

return GlossSynsets(S)
```

Figure 3. The disambiguation algorithm.

Figure 4 shows in dark gray the A-links between *retrospective#1* and the synset of its glosses, while in the light gray area are shown the synsets of the hyperonyms.

3.2 Top-down learning: formal ontologies and WordNet "sweetening"

In the top-down phase, the A-links extracted in the bottom-up phase are refined. A-links are similar to RT (Related Term) relations in thesauri, which provide just a *clue* of relatedness between pairs of thesaurus descriptors[4]. In fact, associations are conceptually ambiguous, since we can only assume that there is *some* relatedness between a synset and another synset extracted from the gloss analysis, but this relatedness must be explicit, in order to understand if it is a hyperonymy relation, or some other conceptual relation (e.g. part, participation, location, etc.).

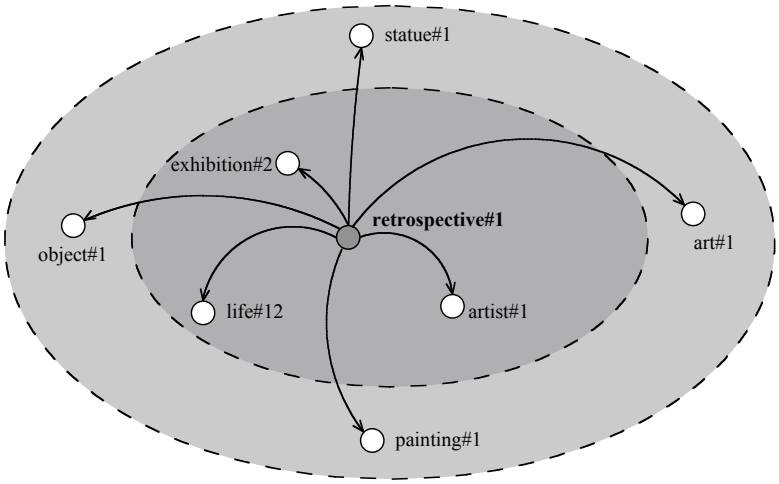


Figure 4. A first approximation of a semantic definition of *retrospective#1*.

First of all, we need a shared set of conceptual relations to be considered as candidates for A-links explicitation, otherwise the result is not easily reusable. Secondly, these relations must be formally defined. In fact, as already pointed out at the beginning of section 3, not only are A-links vague, but they also lack a formal semantics: for example, if we decide (which seems reasonable) to represent associations as binary relations –like DAML+OIL "properties"– is an association symmetric? Does it hold for every instance, or only for some of the instances of the classes derived from the associated synsets? Is it just a constraint on the applicability of a relation to that pair of classes? Is the relation set a flat list, or there is a taxonomic ordering?
To answer such questions, the shared set of relations should be defined in a logical language using a formal semantics.
Since WordNet is a general-purpose resource, the formal shared set of relations should also be general enough, based on *domain-independent* principles, but still flexible, in order to be easily maintained and negotiated.

[4] A-links have an advantage over RT relations, because A-links are directed, while RT are symmetric relations. A-links are directed because we assume that the links hold from a source synset to a synset extracted from its gloss.

3.2.1 The DOLCE descriptive ontology

A proposal in this direction is provided by the WonderWeb[5] project Foundational Ontology Library (WFOL), which will contain a library including both compatible and alternative modules including domain-independent concepts and relations. A recently defined module that accomplishes the abovementioned requirements is DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering).
DOLCE is expressed in an S5 modal logic (Masolo et al. 2002), and has counterparts in computational logics, such as KIF, LOOM, RACER, DAML+OIL, and OWL. The non-KIF counterparts implement a reduced axiomatization of DOLCE, called DOLCE-Lite. DOLCE-Lite has been extended with some *generic plugins* for representing information, communication, plans, ordinary places, and with some *domain plugins* for representing e.g. legal, tourism, biomedical notions. The combination of DOLCE-Lite and the existing plugins is called DOLCE-Lite+. The current version 3.6 of DOLCE-Lite+ without domain plugins contains more than 300 concepts and about 150 relations (see table 1 and 2 in Appendix 1).
DOLCE assumes that its categories (top classes) constitute an *extensionally* closed set on any possible *particular* entity, i.e., entities that cannot be further instantiated within the assumptions of the theory (cf. Masolo et al. 2002, Gangemi et al. 2001). Of course, DOLCE does not assume an *intensionally* closed set, thus allowing for alternative ontologies to co-exist. Such assumptions will be referred to as *A6_D* ("extensional total coverage of DOLCE"). Consequently, we also assume that WN globally can be tentatively considered a (extensional) subset of DOLCE, after its formalization. Since we cannot practically obtain a complete formalization of WN, we will be content with incrementally approximating it.
A trivial formalization of WN might consist in declaring formal subsumptions for all *unique beginners* (top level synsets) under DOLCE categories, but this proved to be impossible, since the intension of unique beginners, once they are formalized as classes, is not consistent with the intension of DOLCE categories. Then we started (Gangemi et al. 2002) deepening our analysis of WN synsets, in order to find synsets that could be subsumed by a DOLCE category (or one of their subclasses) without being inconsistent.
In our previous OntoWordNet work, WordNet 1.6 has been analyzed, and 809 synsets have been relinked to DOLCE-Lite+ in order to harmonize ("sweeten") WN taxonomies with DOLCE-Lite+. A working hypothesis (*A7_D*) has been that the taxonomy branches of the relinked synsets are ontologically consistent with the DOLCE-Lite+ concepts, to which the relinking is targeted. This hypothesis proved inadequate in the initial attempts to get a complete DOLCE coverage of WordNet, since the intended meanings of hyponym synsets are usually not consistent through the entire branching (cf. Gangemi et al. 2002 for examples) After some additional work, the current linking of 809 synsets seems acceptable, but it needs refinement, since some subsumptions are debatable, and it must be considered that some extensions of DOLCE-Lite+ are still unstable.
Nonetheless, such an approximate and partly debatable coverage could be enough to start experimenting with a more explicit axiomatization of synsets. We will show in

[5] http://wonderweb.semanticweb.org

what follows that this experiment can also provide feedback to refine some of the subsumptions.

3.2.2 Disambiguation of association links

Assumptions A4 and A5 (section 2), together with A6_D (in previous sub-section), make it possible to exploit the axiomatized relations in DOLCE-Lite+. Such relations are formally characterized by means of *ground axioms* (e.g. symmetry, transitivity, etc.), *argument restrictions* (qualification of their *universe*), *existential axioms, links to other primitives, theorems,* etc. (refer to (Masolo et al. 2002), and the web site of the LOA).

By looking at the A-links, a human expert can easily decide which relation from DOLCE-Lite+ is applicable in order to disambiguate the A-link, for example, from:

1. A-link(*car#1*, *engine#1*)

we may be able to infer that cars have engines as components:

$\exists x.\ \mathrm{Car}(x) \rightarrow \exists y.\ \mathrm{Engine}(y) \wedge \mathrm{Component}(x,y)$

or that from

2. A-link(*art_exhibition#1*, *painting#1*)

we can infer that exhibitions as collections have paintings as members:

$\exists x.\ \mathrm{Art\_exhibition}(x) \rightarrow \exists y.\ \mathrm{Painting}(y) \wedge \mathrm{Member}(x,y)$

But this is an intellectual technique that requires a lot of effort. We are instead interested, at least for the sake of bootstrapping a preliminary axiomatization of synsets, in a (semi) *automatic classification technique*.
From this viewpoint, the only available structure is represented by the concepts (synsets) to which the A-links apply. Such synsets can be assumed as the *argument restrictions* of a conceptual relation implicit in the association. For example, given (A-link(S_1, S_2)), where S_1, S_2 are synsets, we can introduce the argument restrictions for a conceptual relation $\mathrm{R}^{\text{a-link}}_{\mathrm{i}}(x,y) \rightarrow \mathrm{S}_1(x) \wedge \mathrm{S}_2(y)$. Then, from A5 and its depend-on assumptions, we have a good heuristics for concluding that $\mathrm{S}_1(x) \rightarrow \exists y.\ \mathrm{R}^{\text{a-link}}_{\mathrm{i}}(x,y) \wedge \mathrm{S}_2(y)$. In other words, we formalize the association existing between a synset and another synset used in its gloss. This leaves us with the question of what is the intension of $\mathrm{R}^{\text{a-link}}_{\mathrm{i}}(x,y)$, beyond its argument restrictions: e.g. what does it mean to be a relation between *art exhibitions* and *paintings*? And are we allowed to use this heuristics to conclude that art exhibitions are related to at least one painting?
Assuming A6_D, we can claim that some $\mathrm{R}_{\mathrm{i}}(x,y)$ from DOLCE-Lite+ subsumes $\mathrm{R}^{\text{a-link}}_{\mathrm{i}}(x,y)$. Since the relations from DOLCE-Lite+ have a total extensional coverage on any domain, we can expect that at least one relation from DOLCE has a universe

subsuming that of $R^{a\text{-}link}_i(x,y)$. For example: Member(x,y) from DOLCE-Lite+ can subsume $R^{a\text{-}link}_i(x,y)$ when Art_exhibition(x) and Painting(y), since the domain and range of "Member" subsume "Art_exhibition" and "Painting" respectively.
These subsumptions are easily derivable by using a description-logic classifier (e.g. LOOM, MacGregor, 1993, or RACER, Moeller, 2001) that computes the applicable relations from DOLCE-Lite+ to the training set of A-links.
For example, an "ABox" query like the following can do the job in LOOM:

ABox-1
(retrieve (?x ?R ?y) (and (get-role-types ?x ?R ?y) (min-cardinality ?x ?R 1) (A-link ?x ?y)))

i.e., provided that A-links have been defined on DOLCE-Lite+ classes (i.e. that WN synsets ?x ?y are subsumed by DOLCE-Lite+ classes), the relation "get-role-types" will hold for all the relations in DOLCE-Lite+ that are applicable to those classes, with a cardinality≥1. For example, given the previous example (2) of A-link, the classifier uses some of the DOLCE-Lite+ axioms to suggest the right conceptual relation. In fact, the WordNet synset *art_exhibition#1* is a (indirect) sub-class of the DOLCE class "unitary collection", a category for which the following axiom holds:

$$\forall x.\ \text{Unitary\_Collection}(x) \rightarrow \exists y.\ \text{Physical\_Object}(y) \wedge \text{Member}(x,y)$$

Furthermore, since *painting#1* is a (indirect) sub-class of "physical object", and the axiom holds with a cardinality≥1, the classifier can propose the correct relation and axiom.

In other cases, ABox-1 retrieves relations that are questionable. For example, given:

3. A-link(*boat#1*,*travel#1*)

with *boat#1* subsumed by Physical_Object and *travel#1* subsumed by Situation in DOLCE+WordNet, and the relation "Setting" holding between physical objects and situations, we have no axiom like the following in DOLCE-Lite+:

$$*\ \forall x.\ \text{Physical\_Object}(x) \rightarrow \exists y.\ \text{Situation}(y) \wedge \text{Setting}(x,y)$$

then the relation $R^{a\text{-}link}_i$ formalizing the A-link between *boat* and *travel* cannot be automatically classified and proposed as subsumed by the relation "Setting" in DOLCE-Lite+. In other words, in general *it is not true* that "for any physical object there is at least a situation as its possible "*setting*": we can figure out physical objects in general, without setting them anywhere, at least within the scope of a computational ontology.
In other cases, there exists a potentially appropriate relation, but it is applied in an incorrect way. For example, given:

4. A-link(motor hotel#1,parking area#1)

DOLCE-Lite+ provides the relation "*spatial-location"*, holding between objects and regions. According to its argument restrictions, DOLCE-Lite+ suggests that *motor hotel* (subsumed by *object*) *is located* in a *parking area* (subsumed by *space region*). But it is imprecise: actually, the parking area is located in the overall area of the motor hotel.

The above examples show that axioms representing generally acceptable intuitions in a foundational ontology may prove inadequate in a given application domain, where certain axiomatizations need an ad-hoc refinement.

The solution presented here exploits a partition of argument restrictions for the gloss axiomatization task. For this solution, we need a partition □ of relation universes, according to the 25 valid pairs of argument restrictions that can be generated out of the five top categories of DOLCE-Lite+ (*Object*, *Event*, *Quality*, *Region*, and *Situation*), which on their turn constitute a partition on the domain of entities for DOLCE-Lite+. This enables us to assign one of the 25 relations to the A-link whose members are subsumed by the domain and range of that relation. For example, from: (Boat(x) □ Object(x)), and (Travel(y) □ Situation(y)), we can infer that some $R_{<Object,Situation>}$ holds for the pair $\{x,y\}$.

However, in DOLCE-Lite+, existing relations are based on primitives adapted from the literature, covering some basic intuitions and that are axiomatized accordingly. Therefore, the current set of DOLCE-Lite+ relations □□ is not isomorphic with □, while the same extensional coverage is supported. For example, the DOLCE-Lite+ relation "part" corresponds to a *subset* of the union of *all* the argument pairs in □ that include only the same category (e.g., <Event, Event>). □□ is inadequate to perform an automatic learning of conceptual relations, because we cannot distinguish between "part" and other relations with the same universe (e.g. "connection"). Similarly, we cannot distinguish between different pairs of argument restrictions *within* the "part" universe (e.g. <Event, Event> vs. <Object, Object>).

The choice of axioms in DOLCE-Lite+ is motivated by the necessity of *grounding* the primitive relations in human intuition, for example in so-called *cognitive schemata* that are established during the first steps of an organism's life by interacting with its environment and using its specific abilities to react to the stimuli, constraints, and affordances provided by the context (Johnson 1987). In fact, without that grounding, the meaning of relations cannot be figured out at all (even though they are correct from a logical viewpoint).

There is also another reason for the inadequacy of □□. A conceptual relation in DOLCE-Lite+ can be "mediated", e.g. defined through a *composition* (called also *chaining*, or *joining* in the database domain). For example, two objects can be related because they participate in a same event, for example, *engine* and *driver* can "co-participate" because they both *participate in driving*.

In brief: we cannot use □□, since it does not discriminate at the necessary level of detail, and because it is not a partition at all, if we take into account mediated relations. On the other hand, we cannot use □ , because it is cognitively inadequate.

Consequently, we have evolved a special partition □□+ that keeps both worlds: a real partition, and cognitive adequacy. □□+ denotes a partition with a precise mapping to □□. In appendix 2, the current state of □□+ is shown.

For example, by using DOLCE-Lite+, the proposed relation for the *car*/*engine* example is *(Physical-)Mereotopological-Association* (PMA), defined as the union of some DOLCE-Lite+ primitive relations: part, connection, localization, constituency, etc., holding only within the *physical object* category. In fact, many possible relational paths can be walked from an instance of *physical object* to another, and only a wide-scope relation can cover them all. Formally:

$$\begin{aligned}\mathrm{PMA}(x,y) =_{df} (&\mathrm{Part}(x,y) \cup \mathrm{Overlaps}(x,y) \cup \mathrm{Strong\text{-}Connection}(x,y) \\ &\mathrm{Weak\text{-}Connection}(x,y) \cup \mathrm{Successor}(x,y) \cup \mathrm{Constituent}(x,y) \\ &\mathrm{Approximate\text{-}Location}(x,y)) \cap \\ &\cap \mathrm{Physical\_Object}(x) \cap \mathrm{Physical\_Object}(y)\end{aligned}$$

Starting from DOLCE-Lite+, other relations have been defined for subsets of the domains and ranges of the relations in DOLCE-Lite+.
By means of DOLCE-Lite+, the query function ABox-1 can be adjusted as follows:

ABox-2

```
(retrieve (?x ?r ?y)
 (and
  (A-Link ?x ?y)
  (Superrelations ?x Physical_Object)
  (Superrelations ?y Physical_Object)
  (not
   (and (Superrelations?x Unitary_Collection)
        (Superrelations?y Physical_Object)))
  (not
   (and (Superrelations?x Amount_of_Matter)
        (Superrelations?y Physical_Body)))
  (not (subject ?x dolce))
  (not (subject ?y dolce))
  (not (Superrelations ?x ?y))
  (not (Superrelations ?y ?x))
  (min-cardinality ?x ?r 1)))
```

The query approximately reads “if two synsets subsumed by *physical object* (provided that the first is not an amount of matter or a collection, and that they are not related by hyperonymy), are linked by an A-link, tell me what relations in DOLCE+WordNet are applicable between those synsets with a cardinality of at least 1”.

In this way, we are able to learn all the relations that are applicable to the classes ?x and ?y involved in the A-Link tuples. The intention here is, for example, to limit the universe of “PMA”, in order to give room to more specific relations, such as “Member” or “Constituent”, with specialized universes. For example, applied to the synset *car#1* that has an A-link to the synset *engine#1*, the query returns:

$$R_{PMA}(car\#1, engine\#1)$$

that, on the basis of known assumptions, is used to propose an axiom on *car#1*, stating that cars have a “physical mereotopological association” with an *engine*,

because a DOLCE-Lite+ ancestor of both *car#1* and *engine#1* ("*physical object*") defines the universe of the relation PMA with a cardinality of at least 1 on the range. This heuristics supports the logical axiom:

$$\forall x.\ \mathrm{Car}(x) \rightarrow \exists y.\ \mathrm{Engine}(y) \wedge \mathrm{PMA}(x,y)$$

Notice that at this level of generality, the classifier cannot infer the "component" relation that we intellectually guessed at the beginning of section 3.2. A more specific relation can be approximated, if we define more specialised relations and axioms. For example, a "functional co-participation" can be defined with a universe of only "functional objects", which are lower in the DOLCE-Lite+ taxonomy, but still higher than the pair of synsets associated by the A-link. Functional co-participation ("FCP") is defined by composing two participation relations with a common event (in the example, a common event could be "car running"):

$$\mathrm{FCP}(x,y) =_{\mathrm{df}} \exists z.\ \mathrm{Participant\_in}(x,z) \wedge \mathrm{Participant}(y,z) \wedge \mathrm{Event}(z)$$

FCP is closer to the "component" intuition. The last can be precisely inferred if we feed the classifier with "core" domain relations. For example, we may define a domain relation holding for vehicles and functional objects, provided that the functional object plays the role of system component for vehicles:

$$\begin{aligned} \textit{vehicles}\text{^}\mathrm{Component}(x,y) =_{\mathrm{df}}\ & \mathrm{FCP}(x,y) \wedge \mathrm{Vehicle}(x) \wedge \mathrm{Functional\_Object}(y) \wedge \\ & \exists z.\ \mathrm{Plays}(y,z) \wedge \mathrm{Vehicle\_System\_Component}(z) \end{aligned}$$

In other words, by increasing the specificity of the domain (tourism in the examples discussed so far), we may assume that relations should be specified accordingly. As discussed in this section, this process is triggered by the observation of some A-link, and proceeds semi-automatically until a reasonable coverage is reached.
Anyway, when the domain cannot be specified, even a generic association like "PMA" provides a better intuition than a bare A-link.
The conceptual relation partition is being incrementally verified, and the results of the experiment presented here can also be used as a test bed for creating a pruned set of *domain-oriented* relations. Notice that the pruned set of relations $\mathcal{DOLCE}$+ is always consistent with the original DOLCE-Lite+ conceptual relations, with which the pruned relations form a larger intensional set (the extensional coverage is maintained).

4. Experimental results and discussion

The gloss disambiguation algorithm and the A-link interpretation methods have been evaluated on two sets of glosses: a first set of 100 general-purpose glosses[6] and a

[6] The 100 generic glosses have been randomly selected among the 809 glosses used to re-link WordNet to DOLCE-Lite+.

second set of 305 glosses from a tourism domain. This allows us to evaluate the method both on a restricted domain and a non-specialized task.
For each term in a gloss, the appropriate WordNet sense has been manually assigned by two annotators, for over 1000 words.
To assess the performance of the gloss disambiguation algorithm we used two common evaluation measures: *recall* and *precision*. Recall provides the percentage of right senses with respect to the overall number of terms contained in the examined glosses. In fact, when the disambiguation algorithm terminates, the list *P* may still include terms for which no relation with the synsets in *D* could be found. Precision measures the percentage of right senses with respect to the retrieved gloss senses. A baseline precision is also computed, using the "first sense choice" heuristic. In WordNet, synsets are ordered by probability of use, i.e. the first synset is the most likely sense. For a fair comparison, the baseline is computed only on the words for which the algorithm could retrieve a synset.

Domains	# glosses	# words	# disamb. words	# of which ok	Recall	Precision	Baseline Precision
Tourism	305	1345	636	591	47,28%	92,92%	82,55%
Generic	100	421	173	166	41,09%	95,95%	67,05%

Domains	noun recall	noun precision	adj recall	adj precision	verb recall	verb precision	# tot nouns	# tot adj	# tot verbs
Tourism	64,52%	92,86%	28,72%	89,29%	9,18%	77,78%	868	195	294
Generic	58,27%	95,95%	28,38%	95,24%	5,32%	80%	254	74	94

Table 1a) performance of the gloss disambiguation algorithm b) performance by morphological category.

Table 1 gives an overview of the results. Table 1a provides an overall evaluation of the algorithm, while table 1b computes precision and recall grouped by morphological category. The precision is quite high (well over 90% for both general and domain glosses) but the recall is around 40%. Remarkably, the achieved improvement in precision with respect to the baseline is much higher for general glosses than for domain glosses. This is motivated by the fact that general glosses include words that are more ambiguous than those in domain glosses. Therefore, the general gloss baseline is quite low. This means also that the disambiguation task is far more complex in the case of general glosses, where our algorithm shows particularly good performance.
An analysis of performance by morphological category (Table 1b) shows that noun disambiguation has much higher recall and precision. This is motivated by the fact that, in WordNet, noun definitions are richer than for verbs and adjectives. The WordNet hierarchy for verbs is known as being more problematic with respect to nouns. In the future, we plan to integrate in our algorithm verb information from

FRAMENET[7], a lexico-semantic knowledge base providing rich information especially for verbs.
In Table 2 we summarize the efficacy of the A-link semi-automatic axiomatization, after the partly manual creation of a domain view $\mathcal{DL}^+$ as discussed in section 3.2.

Domains	Synsets	A-links	Noun-only	Subsumptions	Filtered A-links	Axioms generated	Correct
Tourism	305	725	644	209	435	569	511
Generic	100	212	187	40	147	142	121

Table 2. Axiomatizations for the A-links. "Best arrangement" data refer to results in Table 3.

	Tourism	Tourism correct	Generic	Generic correct
Total amount of axioms	569	511 (89.80%)	142	121 (85.21%)
Axioms with generic universes	540	490 (90.74%)	139	121 (87.05%)
Axioms with some specific universes	545	507 (93.02%)	136	118 (86.76%)
Axioms with only topmost universes	375	356 (94.93%)	110	98 (89.09%)

Table 3. Axiomatizations ordered by generality.

As a preventive measure, we have excluded the A-links that include either an adjective or a verb, since these synsets have not been integrated yet with DOLCE-Lite+. Another measure excluded the A-links that imply a subsumption (sub-class) link, since these are already formalized. This filter has been implemented as a simple ABox query that uses relations that range on classes:

ABox-3
(retrieve (?x ?y) (and (A-Link ?x ?y) (Superrelations ?x ?y)))

These measures reduced the amount of A-Links from the experimental set to 582 (435+147). We have used these tuples to run the revised query ABox-2.
The revised query produced 711 (569+142) candidate axioms by using all the pruned relations defined for the experiment in $\mathcal{DL}^+$. Table 3 shows the resulting axioms ordered by generality of the relation universes (domain and range).
The most relevant results are:

[7] http://www.icsi.berkeley.edu/~framenet/

- One third of the A-Links from the tourism domain are actually subsumption links, while only 20% from the mixed generic set is a subsumption. This could be explained by the fact that glosses for generic synsets are less informative, or because generic words are not defined, in WN, in terms of more generic ones.
- The correct subset of axioms learnt for the tourism domain is about 4 to 6% larger than for the generic one with reference to the whole sets.
- We have tried to use some relations that are in principle "less precise". For example, a universe composed of *physical objects* and *amounts of matter* has a basic intuition of "constituency", and the relation *has_n_constituent* has been defined to such purpose. This relation has proved very inefficient though: in the generic set, only 50% of learnt axioms are correct, while in the tourism domain, only 16% are correct. We could expect that domains like *earth science* and *physics* can be more appropriate for constituency relations. For this reason, we have included a relation with a functional flavor in the experimental set of relations (including □+ and its specializations), called "provides", and defined on *functional objects* and *functional matters* (this universe is a meaningful subset of the previous one). This relation proved quite efficient in the tourism domain, just as expected, with about 78% of correct axioms, while it is useless in the generic set, with 0%. This is an example of "provides" axioms: $\forall x$. Brasserie(x) $\rightarrow$ $\exists y$. Beer(y) $\wedge$ Provides(x,y).

  This, and similar examples, confirm our expectations about the importance of developing dedicated sets of relations for different domains or tasks, while a "ground" level of relations is useful everywhere: in fact, the percentage of correct axioms increases if only the first level of the relation hierarchy is taken into account (95% in tourism, 89% in generic).
- In 8 cases, the axioms were not definable with a cardinality≥1, although they could be used in more restricted domains or for subclasses of the universe.
- Some indirect A-links can be investigated as well (though our first strategy has been to disregard indirect links, as explained in section 3.1). For example in the *retrospective#1* example of Figure 2, two synsets (*painting#1* and *statue#1*) are learnt as "indirect" synsets (they are learnt from the glosses relative to the hyperonyms of *retrospective#1*). But paintings and statues are not always found in exhibitions, then we are not allowed to infer an axiom with cardinality ≥1. In these cases, the algorithm could be refined to propose an axiom that includes a common parent to both *painting#1* and *statue#1*, i.e. *art#1*, which incidentally is another "indirect" A-link to *retrospective#1*. In Figure 5 the refined A-links for *retrospective#1* are shown: a *retrospective* in WordNet 1.6 has the intended meaning of a (unitary) collection in DOLCE-Lite+, which is a kind of non-agentive functional object. This lets the classifier infer:

  - a "functional association" to *artist#1*, because an artist is a functional role;
  - a more precise "plays" relation to *life#12*, since an artistic biography is a functional role as well, and a collection of art works plays just the role of an artistic biography;
  - a subsumption of *retrospective#1* by *exhibition#2*;
  - three "has_member" relationships to the indirect A-links: *art#1*, *painting#1*, and *statue#1*. These are correct, since a collection can have functional

objects (art works) as members. But while the first has a meaningful cardinality 1 to n, the others have a logically irrelevant cardinality of 0 to n.

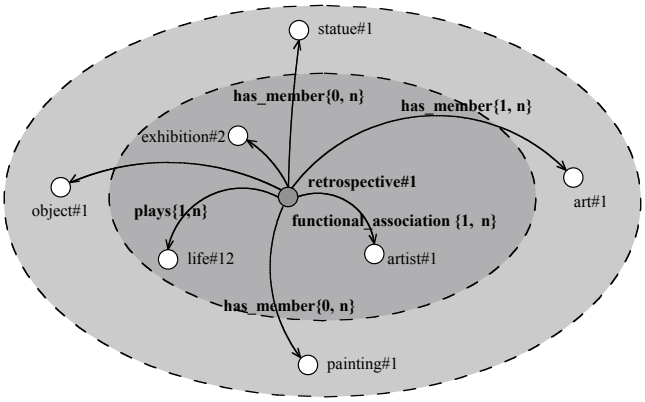


Figure 5. Interpretation of A-links for *retrospective#1*.

Conclusions

In this paper we have presented some preliminary results of OntoWordNet, a large-scale project aiming at the "ontologization" of WordNet. We presented a two step methodology: during the first, automatic phase, natural language word sense glosses in WordNet are parsed, generating a first, approximate definition of WN concepts (originally called synsets). In this definition, generic associations (A-links) are established between a concept and the concepts that co-occur in its gloss.
In a second phase, the foundational top ontology DOLCE (in the *DOLCE-Lite+* version), including few hundreds formally defined concepts and conceptual relations, is used to interpret A-links in terms of axiomatised conceptual relations. This is a partly automatic technique that involves generating solutions on the basis of the available axioms, and then creating a specialized partition of the axioms (the set □□+ and its specializations) in order to capture more domain-specific phenomena.
Overall, the experiments that we conducted show that a high performance may be obtained through the use of automatic techniques, significantly reducing the manual effort that would be necessary to pursue the objective of the OntoWordNet project.

References

(Basili et al. 1996) Basili R., Pazienza M.T. and Velardi P. *An Empirical Symbolic Approach to Natural Language Processing*, Artificial Intelligence, n. 85, pp.59-99, (1996).

(Berners-Lee, 1998) Berners-Lee T., Semantic Web Road map, http://www.w3.org/DesignIssues/Semantic.html, 1998

(Fellbaum 1995) Fellbaum, C. *WordNet: an electronic lexical database*, Cambridge, MIT press, (1995).

(Gangemi et al. 2001) Gangemi, A., Guarino, N., Masolo, C., and Oltramari, A. 2001. Understanding top-level ontological distinctions. In *Proceedings of IJCAI-01 Workshop on Ontologies and Information Sharing*. Seattle, USA, AAAI Press: 26-33. http://SunSITE.Informatik.RWTHAachen. DE/Publications/CEUR-WS/Vol-47/

(Gangemi et al. 2002) Gangemi A., Guarino N. Masolo C. Oltramari A., Schneider L. "Sweetening Ontologies with DOLCE", Proc. Of EKAW02 http://citeseer.nj.nec.com/cache/papers/cs/26864/http

(Harabagiu and Moldovan 1999) Harabagiu S. and Moldovan D. *Enriching the WordNet Taxonomy with Contextual Knowledge Acquired from Text.* AAAI/MIT Press, (1999).

(Karkaletsis V. Cucchiarelli A Paliouras G. Spyropolous C. Velardi P. "Automatic adaptation of Proper Noun Dictionaries through cooperation of machine learning and probabilistic methods" 23rd annual ACM-SIGIR, Athens, June 2000.

(Johnson 1987) Johnson, Mark. 1987. *The Body in the Mind*. Chicago: University of Chicago Press.

(Mac Gregor, 1993) MacGregor, R. M. 1993. Using a Description Classifier to Enhance Deductive Inference. In *Proceedings of Seventh IEEE Conference on AI Applications*: 141-147.

(Masolo et al. 2002) Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, Alessandro Oltramari, and Luc Schneider. The WonderWeb Library of Foundational Ontologies. WonderWeb Deliverable 17, 2002.

(Magnini and Caviglia 2000) Magnini, B. and Cavaglia, G.: Integrating Subject Field Codes into WordNet. Proceedings of the 2nd International Conference on Language resources and Evaluation, LREC2000, Atenas .

(Miller et al. 1993) G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross & K. Miller; "Introduction to WordNet: An On-Line Lexical Database"; http://www.cosgi.princeton.edu/~wn; August 1993.

(Mihalcea and Moldovan, 2001) Milhalcea, R. and Moldovan. D. *eXtended WordNet: progress report*. NAACL 2001 Workshop on WordNet and Other Lexical Resources, Pittsbourgh, June (2001).

(Moeller, 2001) Volker Haarslev, Ralf Möller Description of the RACER System and its Applications Proceedings International Workshop on Description Logics (DL-2001), Stanford, USA, 1.-3. August 2001

(Morin, 1999) Morin E., *Automatic Acquisition of semantic relations between terms from technical corpora*, Proc. of 5th International Congress on Terminology and Knowledge extraction, TKE-99, (1999).

(Navigli et al. 2003) R. Navigli, P. Velardi and A. Gangemi, *Ontology Learning and its Application to Automated Terminology Translation*, IEEE Intelligent Systems, vol. 18, n.1, pp. 22-31, January 2003

(Searle, 1985) Searle, J.R and Vanderveken, D. *Foundations of Illocutionary Logics*, Cambridge UP, (1985).

(Smith and Welty, 2001) Smith, B. and Welty, C. *Ontology: towards a new synthesis*, Formal Ontology in Information Systems, ACM Press, (2001).

(Vossen 2001) Vossen P. *Extending, Trimming and Fusing WordNet for technical Documents*, NAACL 2001 workshop on WordNet and Other Lexical Resources, Pittsburgh, July (2001).

Web Sites

DAML+OIL http://www.daml.org/2001/03/daml+oil-index

LDC corpus http://www.ldc.upenn.edu/

FRAMENET http://www.icsi.berkeley.edu/~framenet/

WordNet 1.6 http://www.cogsci.princeton.edu/~wn/w3wn.html

Semcor http://engr.smu.edu/~rada/semcor/

WonderWeb http://wonderweb.semanticweb.org

Appendix 1

DOLCE-Lite+ top taxonomies of classes and (binary) relations, presented in hierarchical form, with short descriptions and examples.

Entity	*Anything conceived as no more instantiatable*
: Quality-Space	*A space of values (e.g. dimensional spaces)*
: : Region	*A value or range of values in a (dimensional) space*
: : : Abstract-Region	*Non-physical or temporal values (e.g. monetary)*
: : : Physical-Region	*Physical values (e.g. volume, color, geographic space)*
: : : Temporal-Region	*Temporal values (e.g. gregorian date system)*
: Quality	*An individual counterpart of a region (e.g. the color of a rose)*
: Endurant (≈Object)	*An entity with a direct spatial value (localization)*
: : Non-Physical-Endurant	*A non-physical object, such as social or mental objects*
: : : S-Description	*A reified conceptualization or theory (e.g. plans, norms)*
: : : Course	*An (abstract) sequence of activities (cf. process model)*
: : : Functional-Role	*A role played by an object (e.g. minister, student)*
: : : Parameter	*A selection of value sets (e.g. speed limit)*
: : Physical-Endurant (≈Substance)	*A physical entity with a direct localization, wholly present at a snapshot, cf. Substance*
: : : Amount-Of-Matter	*An amount of matter without a unity (e.g. sand, milk)*
: : : : Functional-Matter	*An amount of matter according to scope (e.g. food)*
: : : Feature	*A relevant part within an object (e.g. edges, holes)*
: : : Physical-Object	*A substance with a unity criterion (e.g. stones, roses)*
: : : : Agentive-Physical-Object	*A physical object with intentionality (e.g. organisms, robots)*
: : : : : Agentive-Functional-Object	*An agentive object according to some scope (e.g. robots)*
: : : : Non-Agentive-Physical-Object	*A physical object without intentionality (e.g. stones, livers)*

: : : : : Non-Agentive-Functional-Object	*A non-agentive object according to some scope (e.g. hammers, walls)*
: Perdurant (≈Event, Process)	*An entity with a direct temporal value (temporal presence), present only as spanning through time*
: : Event	*A temporal entity with heterogeneous parts (e.g. activities, phenomena)*
: : State	*A temporal entity with homogeneous parts*
: Situation	*A reified model or structure (e.g. conditions, environments, states of affairs, observed facts)*

Conceptual-Relation	*Entity(x), Entity(y). The top-level relation between entities whatsoever.*
: Immediate-Relation	*Any relation holding directly, without any other intermediate relation chaining*
: : Constituent	*A relation of constituency between e.g. matter and objects, e.g. skin made up of epithelial tissue*
: : : Has-Member	*A constituency between collections and their members, e.g. a society and its members*
: : : Setting-For	*A constituency between situations and their entities, e.g. a flu and its observed symptoms*
: : Host	*Feature(x), Physical-Endurant(y). The relation between features and objects, e.g. a hole in the cheese*
: : Inherent-In	*Quality(x), Entity(y). The relation between qualities and entities, e.g. the red of a rose*
: : Part	*Any part relation (but not constituency), e.g. a chair and its legs*
: : : Proper-Part	*Any antisymmetric part, e.g. a human body and its legs*
: : : : Boundary	*A part relation between an entity and its boundary, e.g. Italy's borders*
: : : : Component	*A functional, non-transitive part relation, e.g. a car and its parts*
: : Participant	*Event(x), Object(y), the relation for taking part in something, e.g. love and lovers*
: : Q-Location	*Quality(x), Quality-Space(y), the relation between qualities and their counterparts, e.g. the red of a rose and its representation in a color palette*
: : References	*S-description(x), Situation(y), the relation between conceptualizations and situations, e.g. a plan and an activity executed according to that plan*
: : : Played-By	*Functional-role(x), Object(y), the relation for role-playing, e.g. student and a person who is enlisted in a university*
: : Weak-Connection	*A generic, unordered connection*
: : Predecessor	*An ordered connection, e.g. between two consecutive intervals*
: Mediated-Relation	*Any relation holding indirectly, for which some other relation must hold preliminarily*
: : Co-Partipation	*Object(x), Object(y), the relation holding between two objects that participate in th same event or state*
: : Generic-Location	*Any location relation between entities whatsoever*
: : : Exact-Location	*Any location between objects or events, and a) region, e.g. Rome and its geographic coordinates, a stone and its volume*
: : : Approximate-Location	*Any location between entities other than regions, e.g. the pen is on the table*

Appendix 2

The experimental set of relations (/ + and its specializations, argument restrictions into brackets). Only retrieved relations are listed, with their numerosity in the experimental glosses, and the amount of correct assignments.

Relation taxonomy	**Tourism**	***Tou. corr.***	**Generic**	***Gen. corr.***
Conceptual_Relation (Entity, Entity)	*top: correct by A5*			
: Descriptive_Association (Object, S-Description)	7	*6*	5	*4*
: : Descriptive_Constituent_Of (Functional-Role, S-Description)			1	*0*
: Inv_Descriptive_Association (S-Description, Object)	7	*7*	4	*4*
: : Has_Descriptive_Constituent (S-description, Functional-Role)	1	*0*		
: Functional_Association (Object, Functional-Role)	72	*68*	22	*19*
: : Functional_Role_Co_Participation (F-Role,F-Role)	21	*21*	13	*12*
: Inv_Functional_Association (Object, Functional-Role)	45	*45*	21	*19*
: Physical_Location_Of (Geographical-Entity, Physical-Object)	2	*2*	2	*2*
: : Functional_Location_Of (Geographical-Entity, Functional-Object)	1	*1*		
: Has_Physical_Location (Physical-Object, Geographical-Entity)	6	*3*		
: : Has_Functional_Location (Functional-Object, Geographical-Entity)	6	*3*		
: Quality_Region_Of (Region, Object)	3	*3*	2	*1*
: Has_Quality_Region (Object, Region)	9	*8*	2	*0*
: Host_Of (Physical-Object, Feature)	7	*2*	3	*2*
: Host (Feature, Physical-Object)	1	*1*		
: Mereotopological_Association (Physical-Object, Physical-Object)	140	*140*	29	*29*
: : Agentive_CoParticipation (Agentive-Physical-Object, A.P.O.)	1	*1*	2	*1*
: : Functional_CoParticipation (Functional-Object, Functional-Object)	98	*94*	1	*1*
: : Has_Member (Collection, Object)	4	*4*		
: : Provides (Functional-Object, Functional-Matter)	22	*17*	3	*0*
: : Biological_Part_Of (Biological-Object, Organism)			4	*4*
: : Has_Material_Constituent (Physical-Object, Amount-Of-Matter)	24	*4*	6	*3*
: : Used_By_Co_Pcp (Functional-Object, Agentive-Physical-Object)	7	*4*		
: : Member_Of (Object, Collection)	1	*0*		
: Participant (Event, Object)	14	*14*		
: : Agentive_Participant (Event, Agentive-Object)	3	*3*		
: Participant_In (Object, Event)	14	*13*	6	*6*
: : Agentive_Participant_In (Functional-Object, Event)	1	*1*		
: P_Has_Quality_Region (Event, Region)	1	*1*		
: Setting_For (Situation, Entity)	18	*17*		
: : Referenced_By (Situation, S-Description)	1	*0*		
: Setting (Entity, Situation)	21	*21*	8	*7*
: : References (S-Description, Situation)	3	*2*	2	*2*
: Temporal_ Mereotopological_Association (Event, Event)	6	*5*	2	*1*
: Inherence_Of (Entity, Quality)	2	*0*	4	*4*

Automatic Extraction of Knowledge from Web Documents

Harith Alani, Sanghee Kim, David E. Millard, Mark J. Weal,
Paul H. Lewis, Wendy Hall, Nigel Shadbolt

I.A.M. Group, ECS Dept.
University of Southampton
Southampton, UK
{ha, sk, dem, mjw, phl, wh, nrs}@ecs.soton.ac.uk

Abstract. A large amount of digital information available is written as text documents in the form of web pages, reports, papers, emails, etc. Extracting the knowledge of interest from such documents from multiple sources in a timely fashion is therefore crucial. This paper provides an update on the Artequakt system which uses natural language tools to automatically extract knowledge about artists from multiple documents based on a predefined ontology. The ontology represents the type and form of knowledge to extract. This knowledge is then used to generate tailored biographies. The information extraction process of Artequakt is detailed and evaluated in this paper.

1 Introduction

Quick analysis and understanding of unstructured text is becoming increasingly important with the huge increase in the number of digital documents available. This has led to an increased use of various tools developed to help levy the problem of processing unstructured text documents through automatic classification, concept recognition, text summarisation, etc.

These tools are often based on traditional natural language techniques, statistical analysis, and machine learning, dealing mostly with single documents. The ability to extract certain types of knowledge from multiple documents and to maintain it in structured Knowledge Bases (KB) for further inference and report generation is a more complex process. This forms the aim of the Artequakt project.

1.1 Relation Extraction

There exist many information extraction (IE) systems that enable the recognition of entities within documents (e.g. ‘Renoir’ is a ‘Person’, ‘25 Feb 1841’ is a ‘Date’). However, such information is incomplete and sometimes insufficient for certain requirements without acquiring the relation between these entities (e.g. ‘Renoir’ was

born on ‘25 Feb 1841’). Extracting such relations automatically is difficult, but crucial to complete the acquisition of knowledge fragments and ontology population (building the KB). The MUC-7 systems [8] are example attempts for extracting a limited number of relations. Whereas the MUC participant systems used training examples to induce a set of rules for named-entity and relation extraction, Artequakt assumes a case where the number and type of relations to be extracted is non-static. Artequakt attempts to identify relations between the entities of interest within sentences, following ontology relation declarations and lexical information.

1.2 Ontology Population

Artequakt is also concerned with automating ontology population with knowledge triples, and providing this knowledge for a biography generation service.

When analysing documents and extracting information, it is inevitable that duplicated and contradictory information will be extracted. Handling such information is challenging for automatic extraction and ontology population approaches [16]. Artequakt applies a set of heuristics and reasoning methods in an attempt to distinguish conflicting information, verify it, and to identify and merge duplicate assertions in the KB automatically

1.3 Biography Generation

Storing information in a structured KB provides the needed infrastructure for a variety of knowledge services. One interesting service is to reconstruct the original source material in new ways, producing a dynamic presentation tailored to the users needs.

Previous work in this area has highlighted the difficulties of maintaining a rhetorical structure across a dynamically assembled sequence [14]. Where dynamic narrative is present it has been based around robust story-schema such as the format of a news programme (a sequence of atomic bulletins) [6].

It is our belief that by building a story-schema layer on top of an ontology we can create dynamic stories within a specific domain. In Artequakt we explore the generation of biographies of artists. Populating the ontology through automatic extraction tools might allow those biographies to be constructed from the vast wealth of information that exists on the World Wide Web, thus bringing together pieces of information from multiple sites into one single repository.

2 Artequakt

The Artequakt project has implemented a system that searches the Web and extracts knowledge about artists, based on an ontology describing that domain, and stores this knowledge in a KB to be used for automatically producing personalised biographies of artists. Artequakt draws from the expertise and experience of three separate

projects; *Sculpteur*[1], *Equator*[2], and *AKT*[3]. The main components of Artequakt are described in the following sections.

Artequakt's architecture comprises of three key areas. The first concerns the knowledge extraction tools used to extract factual information from documents and pass it to the ontology server. The second key area is information management and storage. The information is stored by the ontology server and consolidated into a KB which can be queried via an inference engine. The final area is the narrative generation. The Artequakt server takes requests from a reader via a simple Web interface. The request will include an artist and the style of biography to be generated (chronology, summary, fact sheet, etc.). The server uses story templates to render a narrative from the information stored in the KB using a combination of original text fragments and natural language generation.

The architecture is designed to allow different approaches to information extraction to be incorporated with the ontology acting as a mediation layer between the IE and the KB. Currently we are using textual analysis tools to scrape web pages for knowledge, but with the increasing proliferation of the semantic web, additional tools could be added that take advantage of any semantically augmented pages passing the embedded knowledge through the KB.

2.1 The Artequakt Ontology

For Artequakt the requirement was to build an ontology to represent the domain of artists and artefacts. The main part of this ontology was constructed from selected sections in the CIDOC Conceptual Reference Model (CRM[4]) ontology. The CRM ontology is designed to represent artefacts, their production, ownership, location, etc. This ontology was modified for Artequakt and enriched with additional classes and relationships to represent a variety of information related to artists, their personal information, family relations, relations with other artists, details of their work, etc. The Artequakt ontology and KB are accessible via an ontology server.

3 Knowledge Extraction

The aim of the knowledge extraction tool of Artequakt is to identify and extract knowledge triplets (concept – relation – concept) from text documents and to provide it as XML files for entry into the KB [5]. Artequakt uses an ontology coupled with a general-purpose lexical database (WordNet) [11] and an entity-recogniser (GATE) [3] as supporting tools for identifying knowledge fragments.

---

[1] http://www.sculpteurweb.org/

[2] http://www.equator.ac.uk/

[3] http://www.aktors.org

[4] http://cidoc.ics.forth.gr/index.html

3.1 Document Retrieval

The extraction process is launched when the user requests a biography for a specific artist that is not in the KB. A script was developed to query the artist's name in general-purpose search engines, such as Google and Yahoo.

Documents returned by the search engines need to be filtered to remove irrelevant ones. Expanding queries with additional terms was not very effective for improving the web search. The approach followed in Artequakt is based on query-by-example. In order to pick up pages related to an artist, a short description of the artist from a well-known museum web site (e.g. WebMuseum[5]) is analysed and used as a similarity vector. Structural evidence, such as paragraph length or number of sentences within a paragraph, is used in order to identify and remove pages which mainly consist of links, tables, etc. If the similarity vector is unobtainable (e.g. a search for a relatively new or unknown artist whose entry is not available in the exemplar museum site) the ontology itself is used to create the vector. The quality of entity recognition has a direct effect on the accuracy of relation extraction.

Search for documents stops and the extraction process starts when the number of relevant documents found reaches a specified threshold.

3.2 Entity Recognition

Entity recognition is the first step towards extracting knowledge fragments. GATE is a syntactical pattern matching entity recogniser enriched with gazetteers. GATE's coverage can be expanded with additional extraction rules and gazetteers to enable the identification of further type of entities. However, the process of discovering and setting up new syntactic rules can be difficult and labour intensive. To this end, we deploy WordNet as a supplementary information source in order to identify additional entities not recognised by the default GATE. WordNet is also used to support relation extraction.

3.3 Extraction Procedure

Each selected document is divided into paragraphs and sentences. Each sentence is analysed syntactically and semantically to identify and extract relevant knowledge triples. Figure 1 shows the overall procedure of the extraction process as applied on the sentence:

"Pierre-Auguste Renoir was born in Limoges on February 25, 1841."

[5] http://www.ibiblio.org/wm/

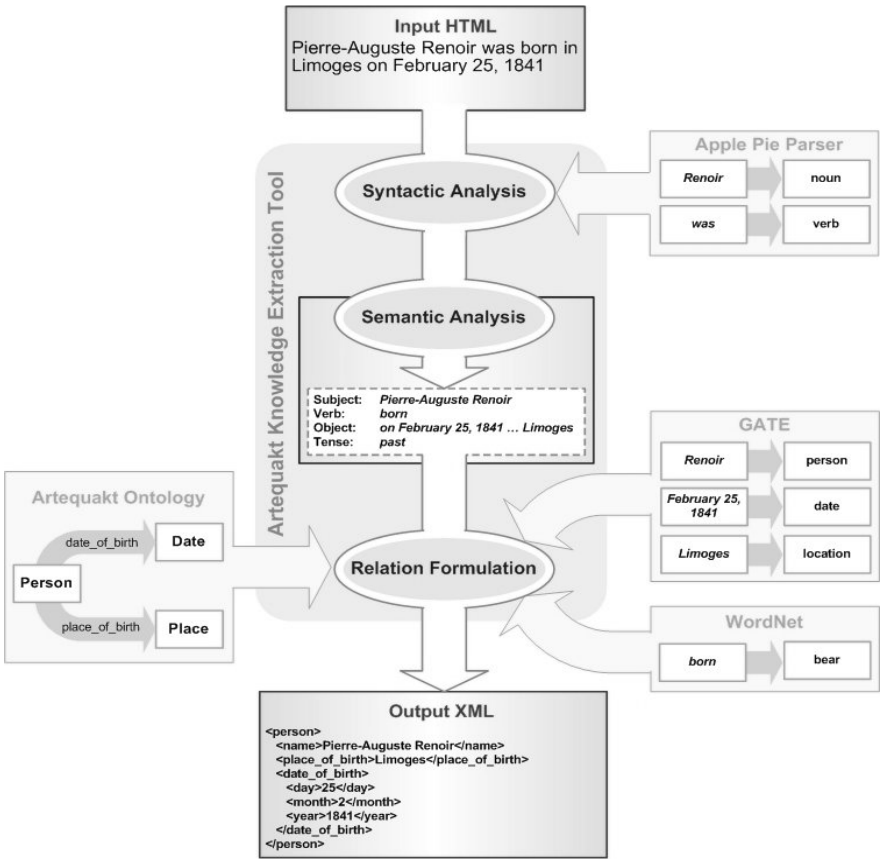


Fig. 1. Artequakt's IE Process

Syntactical Analysis. Syntactical parsing groups words into syntactic functions with no consideration to their semantic meaning. The Apple Pie Parser (APP) [15] is a bottom-up probabilistic chart parser derived from the syntactically tagged corpus; Penn Tree Bank (PTB). PTB contains a large number of example sentences; thus APP tends to have a broad-coverage performance with reasonable accuracy (over 70% for both precision and recall). The output of APP is structured according to the PTB bracketing.

Artequakt makes use of APP to gather syntactical annotations of sentences. For example in Figure 1, APP identified that 'Renoir' is a noun, and 'was' is a verb.

Semantic Analysis. Semantic examination then decomposes the sentence into simple sentences to locate the main components (i.e. subject, verb, object), and identifies named entities (e.g. "Renoir" is a Person, "Paris" is a Location). In the example sentence of Figure 1, "born" is tagged as the main verb in the "was born" verb phrase.

Annotations provided by GATE and WordNet highlight that "Pierre-Auguste Renoir" is a person's name, "February 25, 1841" is a date, and "Limoges" is a

location. GATE is also used to resolve anaphoric references of singular personal pronouns which is crucial for accurate relation extraction.

Term expansion tools are required if the terms identified by the named-entity recogniser differ from those in the ontology. For example, GATE annotates 'Museum of Art' as an Organisation while our ontology defines 'Legal Body' as a general concept for organisations. The system needs to map these two concepts to figure out that 'Museum of Art' is a Legal Body. Currently, we use WordNet for the mapping by looking up the lexical chains of the two terms in search of any overlap.

Relation Extraction. Artequakt is concerned with the extraction of relations between concepts within individual sentences. The aim is to extract relationships between any identified pair of entities within a given sentence. Knowledge about the domain specific semantic relations can be retrieved from the Artequakt ontology to find which relations are expected between the entities in hand.

Relations are extracted by matching the verb and entity pairs found in each sentence with an ontology relation and concept pairs respectively. Three lexical chains (synonyms, hypernyms, and hyponyms) from WordNet are used to expand entity names with related terms to reduce the problem of linguistic variations and increase the chance of matching with other semantically similar terms.

Since a relation may have multiple matchings in WordNet (polysemous words), mapping between a term and an entry in WordNet should into account syntactic and semantic clues present in the given sentence. For example, according to WordNet, 'birth' has four noun senses and one verb sense. The first noun sense is selected since one of its hypernyms is 'time period' which has Date as a hyponym.

For the sentence used in Figure 1, the relation extraction is determined by the categorisation result of the main verb 'bear' which matches with two potential relations in the ontology; 'date_of_birth' and 'place_of_birth'. Since both relations are associated with "February 25, 1841" (Date) and "Limoges" (Place) respectively. After analysing the given sentence, Artequakt generates the following knowledge triples about Renoir:

- `Pierre-Auguste Renoir` *`date_of_birth`* `25/2/1841`
- `Pierre-Auguste Renoir` *`place_of_birth`* `Limoges`

The extraction process terminates by sending the extracted knowledge to the ontology server in XML.

4 Automatic Ontology Population

Storing knowledge extracted from text documents in KBs offers new possibilities for further analysis and reuse. Ontology population refers to the insertion of information into the KB. Populating ontologies with a high quantity and quality of instantiations is one of the main steps towards providing valuable and consistent ontology-based knowledge services. Manual ontology population is very labour intensive and time consuming. A number of semi-automatic approaches have investigated creating document annotations and storing the

results as ontology assertions. MnM [17] and S-CREAM [4] are two example frameworks for user-driven ontology-based annotations, enforced with the IE learning tool; Amilcare [2]. However, these frameworks lack the capability for identifying relationships reliably.

In Artequakt we investigate the possibility of moving towards a fully automatic approach of feeding the ontology with knowledge extracted from unstructured text. Information is extracted in Artequakt with respect to a given ontology and provided as XML files using tags mapped directly from names of classes and relationships in that ontology. When the ontology server receives a new XML file, a *feeder* tool is activated to parse the file and add its knowledge triples to the KB automatically. Once the feeding process terminates, the consolidation tool searches for and merges any duplication in the KB.

Tackling the problem of knowledge integration is important to maintain the referential integrity and quality of results of any ontology-based knowledge service. [16] relied on manually assigned object identifiers to avoid duplication when extracting from multiple documents. Artequakt's knowledge management component attempts to identify inconsistencies and consolidate duplications automatically using a set of heuristics and term expansion methods based on WordNet. Full description of the consolidation procedure is out of the scope of this paper.

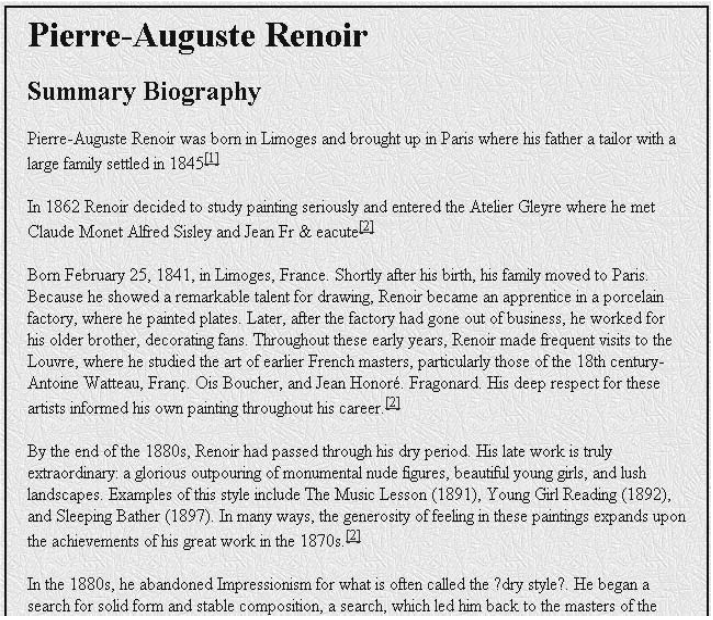
5 Biography Generation

Once the information has been extracted, stored and consolidated, the Artequakt system repurposes it by automatically generating biographies of the artists [1]. The biographies are based on templates authored in the Fundamental Open Hypermedia Model and stored in the Auld Linky contextual structure server [10]. Each section of the template is instantiated with paragraphs or sentences generated from information in the KB.

Different templates can be constructed for different types of biography. Two examples are the summary biography, which provides paragraphs about the artist arranged in a rough chronological order, and the fact sheet, which simply lists a number of facts about the artist, i.e. date of birth, place of study etc. The biographies also take advantage of the structure server's ability to filter the template based on a user's interest. If the reader is not interested in the family life of the artist the biography can be tailored to remove such information. An example of a biography generated by Artequakt can be seen in Figure 2.

By storing conflicting information rather than discarding it during the consolidation process, the opportunity exists to provide biographies that set out arguments as to the facts (with provenance, in the form of links to the original sources) by juxtaposing the conflicting information and allowing the reader to make up their own mind.

As well as searching the KB by name the user interface provides a search facility that allows users to select artists according to other extracted facts, for example the user can specify a range for date of birth and the system will search the appropriate fields with the correct constraints. This kind of query can not be easily formulated over the Web. Extracting the relevant knowledge and storing it in a KB made such queries more feasible.



Pierre-Auguste Renoir

Summary Biography

Pierre-Auguste Renoir was born in Limoges and brought up in Paris where his father a tailor with a large family settled in 1845[1]

In 1862 Renoir decided to study painting seriously and entered the Atelier Gleyre where he met Claude Monet Alfred Sisley and Jean Fr & eacute[2]

Born February 25, 1841, in Limoges, France. Shortly after his birth, his family moved to Paris. Because he showed a remarkable talent for drawing, Renoir became an apprentice in a porcelain factory, where he painted plates. Later, after the factory had gone out of business, he worked for his older brother, decorating fans. Throughout these early years, Renoir made frequent visits to the Louvre, where he studied the art of earlier French masters, particularly those of the 18th century- Antoine Watteau, Franç. Ois Boucher, and Jean Honoré. Fragonard. His deep respect for these artists informed his own painting throughout his career.[2]

By the end of the 1880s, Renoir had passed through his dry period. His late work is truly extraordinary: a glorious outpouring of monumental nude figures, beautiful young girls, and lush landscapes. Examples of this style include The Music Lesson (1891), Young Girl Reading (1892), and Sleeping Bather (1897). In many ways, the generosity of feeling in these paintings expands upon the achievements of his great work in the 1870s.[2]

In the 1880s, he abandoned Impressionism for what is often called the ?dry style?. He began a search for solid form and stable composition, a search, which led him back to the masters of the

Fig. 2. A biography generated using paragraphs.

6 Portability

The use of an ontology to back up IE is aimed to increase the system's portability to other domains. By swapping the current artist ontology with another domain specific one, the IE tool should still be able to function and extract some relevant knowledge, especially if it is concerned with domain independent relations expressed in the ontology, such as personal information. However, certain domain specific extraction rules, such as painting style, will eventually have to be altered to fit the new domain.

Similarly, the generation templates are currently manually tuned to fit biography construction. These templates may need to be modified if a different type of output is required. We aim to investigate developing templates that can be dynamically instructed and modified by the ontology. Building a cross-domain system is one of the ambitions of this project, and will be the focus of the next stage of development.

7 Knowledge Extraction Evaluation

We used the system to populate the KB with information about five artists, extracted from around 50 web pages. Precision and recall were calculated for a set of 10 artist relations (listed in Table 1). The experiment results given in Table 1 shows that precision scored higher that recall with average values of 85 and 42 respectively

Inaccurately extracted knowledge may reduce the quality of the system's output. For this reason, our extraction rules were designed to be of low risk levels to ensure higher extraction precision. Advanced consistency checks could help to identify some extraction inaccuracies; e.g. a date of marriage is before the date of birth, or two unrelated places of birth for the same person!

Table 1. Precision/Recall of extracted relations from around 50 documents for 5 artists

Artist (P/R) / Relation	Rembrandt (P/R)	Renoir (P/R)	Cassatt (P/R)	Goya (P/R)	Courbet (P/R)	Average per relation
Date of birth	75/43	100/50	100/67	80/40	100/100	91/60
Place of birth	100/63	100/14	100/50	100/40	100/63	100/46
Date of death	100/63	100/67	100/50	N/A /0	100/50	100/46
Place of death	100/100	100/43	N/A /0	100/20	100/33	100/49
Place of work	100/50	67/33	33/100	N/A /0	0/0	40/37
Place of study	100/20	100/14	100/75	100/20	100/29	60/32
Date of marriage	100/50	100/33	N/A[1]	100/100	N/A /0	60/46
Name of spouse	100/38	N/A /0	N/A	N/A /0	N/A /0	100/10
Parent profession	100/57	50/67	0/0	67/100	100/100	63/65
Inspired by	100/43	50/60	0/0	100/17	100/33	83/31
Averages	98/53	85/38	61/43	92/34	88/41	**85/42**

The preference of precision versus recall could be dependent on the relation in question. If a relation is of single cardinality, such as a place of birth, then recall could be regarded as less significant as there can only be one value for each occurrence of this relation. A single accurate capture of the value of such a relation could therefore be sufficient for most purposes. However, multiple cardinality relations, such as places where a person worked, can have several values. Higher recall in such cases could be more desirable to ensure capturing multiple values. One possible approach is to automatically adjust the risk level of extraction rules with respect to cardinality, easing the rules if cardinality is high while restricting them further when the cardinality is low.

In Table 1, Goya is an example where few, short documents where found. The amount of knowledge extracted per artist could be used as an automatic trigger to start gathering and analyzing more documents.

8 Related Work

Extracting information from web pages to generate various reports is becoming the focus of much research. The closest work we found to Artequakt is in the area of text

summarisation. A number of summarisation techniques have been developed to help bring together important pieces of information from documents and present them to the user in a compact form. Artequakt differs from such systems in that it aims to extract specific facts and populate a knowledge base with these facts to be used in the generation of personalised reports (e.g. biographies).

Even though most summarisation systems deal with single documents, some have targeted multiple resources [9][18]. Statistical based summarisations tend to be domain independent, but lack the sophistication required for merging information from multiple documents [12]. On the other hand, IE based summarisations are more capable of extracting and merging information from various resources, but due to the use of IE, they are often domain dependent.

Merging information extracted from single or multiple sources is a necessary step towards maintaining the integrity of the extracted knowledge. In many existing IE based systems, information integration is based on linguistics and timeline comparison of single events [12][18] or multiple events [13]. Artequakt's knowledge consolidation is based on the comparison and merging of not just events, but also individual knowledge fragments (e.g. person, place).

Most traditional IE systems are domain dependent due to the use of linguistic rules designed to extract information of specific content, e.g. bombing events (MUC systems), earthquake news [18], sports matches [13]. Adaptive IE systems [2] caould ease this problem by identifying new extraction rules induced from example annotations supplied by users. Using ontologies to back up IE is hoped to support information integration [1][13] and increase domain portability [5][7].

9 Conclusions

This paper describes a system that extracts knowledge automatically, populates an ontology with knowledge triples, and reassembles the knowledge in the form of biographies. Initial experiment using around 50 web pages and 5 artists showed promising results, with nearly 3 thousand unique knowledge triples were extracted, with an average of 85% precision and 42% recall. Preference of precision over recall is subjective and should be associated with relations' cardinality. High precision could be more important for single cardinality relations (e.g. date of birth), while high recall could be preferred for multiple cardinality relations (e.g. places visited).

Future work on Artequakt will continue to develop its modular architecture and refine its information extraction and consolidation processes. In addition we are beginning to look at how we might leverage the full power of the underlying KB and produce biographies that use inference and a dynamic choice of templates to answer a variety of user queries with textual documents. We also intend to investigate the system's portability to other domains.

Acknowledgements

This research is funded in part by EU Framework 5 IST project "Scultpeur" IST-2001-35372, EPSRC IRC project "Equator" GR/N15986/01 and EPSRC IRC project "AKT" GR/N15764/01

References

1. Alani, H., Kim, S., Millard, D., Weal, M., Hall, W., Lewis, P., and Shadbolt, N. "Automatic Ontology-based Knowledge Extraction from Web Documents". *IEEE Intelligent Systems*, 18(1), pages 14-21, 2003.
2. Ciravegna, F. "Adaptive Information Extraction from Text by Rule Induction and Generalisation". *Proc.17th Int. Joint Con. on AI (IJCAI)*, pp 1251--1256, Seattle, USA, 2001.
3. Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. "GATE: a framework and graphical development environment for robust NLP tools and applications". *Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics*, Phil.,USA, 2002.
4. Handschuh, S., Staab, S., and Ciravegna, F. "S-CREAM – Semi Automatic Creation of Metadata". *Semantic Authoring, Annotation and Markup Workshop, 15th European Conf. on Artificial Intelligence*, pages 27--33, Lyon, France, 2002.
5. Kim, S., Alani, H., Hall, W., Lewis, P.H., Millard, D.E., Shadbolt, N., and Weal, M.J. "Artequakt: Generating Tailored Biographies with Automatically Annotated Fragments from the Web". *Workshop on Semantic Authoring, Annotation & Knowledge Markup, 15th European Conf. on Artificial Intelligence (ECAI)*, pages 1--6, Lyon, France, July 2002.
6. Lee, K., D. Luparello, and J. Roudaire, "Automatic Construction of Personalised TV News Programs," *Proc. 7th ACM Conf. on Multimedia*, Orlando, Florida, 1999, pp. 323-332.
7. Maedche, A., G. Neumann and S. Staab. Bootstrapping an Ontology-based Information Extraction System. Intelligent Exploration of the Web. Springer 2002.
8. Marsh, E. & D. Perzanowski (NRL), MUC-7 Evaluation of IE Technology: Overview of Results, available at http://www.itl.nist.gov/iaui/894.02/related_projects/muc/index.html
9. McKeown, K. R., R. Barzilay, D. Evans, V. Hatzivassiloglou, J. L. Klavans, A. Nenkova, C. Sable, B. Schiffman and S. Sigelman. "Tracking and Summarizing News on a Daily Basis with Columbia's Newsblaster". Proc. Human Language Technology Conf., CA, USA. 2002.
10. Michaelides, D.T., Millard, D.E., Weal, M.J., and DeRoure, D. "Auld Leaky: A Contextual Open Hypermedia Link Server". *Proc. of the 7th Hypermedia: Openness, Structural Awareness, and Adaptivity*, pages 59--70, Springer Verlag, Heidelberg, 2001, LNCS.
11. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. "Introduction to wordnet: An on-line lexical database". *Int. Journal of Lexicography*, 3(4):235--312, 1993.
12. Radev, D. R. and K. R. McKeown. "Generating natural language summaries from multiple on-line sources." Compuutational Linguistics 24(3): 469—500, 1998.
13. Reidsma, D., J. Kuper, T. Declerck, H. Saggion and H. Cunningham. Cross document annotation for multimedia retrieval. EACL Workshop on Language Technology and the Semantic Web (NLPXML), Budapest, Hungary, 2003.
14. Rutledge, L., B. Bailey, J.V. Ossenbruggen, L. Hardman, and J. Geurts, "Generating Presentation Constraints from Rhetorical Structure," *Proc. 11th ACM Conf. on Hypertext and Hypermedia*, San Antonio, Texas, USA, 2000, pp. 19-28.
15. Sekine, S. and Grishman R., "A corpus-based probabilistic grammar with only two non-terminals", Proc. of the 1st Int. Workshop on Multimedia annotation, Japan, 2001.
16. Staab, S., Maedche, A., and Handschuh, S. "An Annotation Framework for the Semantic Web". Proc. 1st Int. Workshop on MultiMedia Annotation, Tokoyo, Japan, January 2001.
17. Vargas-Vera, M., E. Motta, J. Domingue, M. Lanzoni, A. Stutt and F. Ciravegna. "MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup". 13th Int. Conf on Knowledge Engineering and Management (EKAW 02), Spain, 2002.
18. White, M., T. Korelsky, C. Cardie, V. Ng, D. Pierce and K. Wagstaff. Multidocument Summarization via Information Extraction. Proc. of Human Language Technology Conf. (HLT 2001), San Diego, CA, 2001.

The Semantic Web: A New Opportunity and Challenge for Human Language Technology

Kalina Bontcheva and Hamish Cunningham

Department of Computer Science, University of Sheffield
211 Portobello St, Sheffield, UK S1 4DP
{kalina,hamish}@dcs.shef.ac.uk
http://gate.ac.uk

Abstract. This position paper motivates the need for Semantic Web enabled Human Language Technology (HLT) tools and discusses the major outstanding challenges in this area. It introduces the idea of a "language loop" and shows how HLT can be used to bridge the gap between the current web of language and the Semantic Web. We also argue for a closer integration between HLT and Semantic Web tools and infrastructures. These challenges are at the core of the research agenda of the upcoming EU-funded SEKT project[1].

1 Introduction

The Semantic Web aims to add a machine tractable, re-purposeable layer to compliment the existing web of natural language hypertext. In order to realise this vision, the creation of semantic annotation, the linking of web pages to ontologies, and the creation, evolution and interrelation of ontologies must become automatic or semi-automatic processes.

In the context of new work on distributed computation, Semantic Web Services (SWSs) go beyond current services by adding ontologies and formal knowledge to support description, discovery, negotiation, mediation and composition. This formal knowledge is often strongly related to informal materials. For example, a service for multi-media content delivery over broadband networks might incorporate conceptual indices of the content, so that a smart VCR (such as next generation TiVO) can reason about programmes to suggest to its owner. Alternatively, a service for B2B catalogue publication has to translate between existing semi-structured catalogues and the more formal catalogues required for SWS purposes. To make these types of services cost-effective we need automatic knowledge harvesting from all forms of content that contain natural language text or spoken data.

[1] http://sekt.semanticweb.org. The SEKT partners are: British Telecommunications Plc.; Empolis GmbH; University of Sheffield; University of Karlsruhe; Jozef Stefan Institute; Institut fur Informatik der Universitat Innsbruck; Intelligent Software Components S. A.; Kea-pro GmbH; Ontoprise GmbH; Sirma AI Ltd; Vrije Universiteit Amsterdam; Autonomous University of Barcelona.

Other services do not have this close connection with informal content, or will be created from scratch using Semantic Web authoring tools. For example, printing or compute cycle or storage services. In these cases the opposite need is present: to document services for the human reader using natural language generation.

Finally, tools and infrastructures for the Semantic Web on the one hand and language technology on the other have so far remained largely independent from each other, despite the fact that they share a number of components, namely ontologies and reasoning mechanisms. HLT systems can benefit from new developments like the Ontology Middleware Module (OMM – an extension of the SESAME RDF(S) repository, see [9]) which will enable HLT tools to index and retrieve language data like annotations and gazetteers in RDF(S). It will also enable the use of Semantic Web reasoning tools within HLT components.

To summarise, recent developments in the Semantic Web field have created new opportunities and challenges for Human Language Technology.

In this position paper we discuss the role of HLT in closing the language loop, provide brief overviews of state-of-the-art approaches to tackling some aspects of the problem, and discuss a number of open issues that remain to be solved. The paper is organised as follows. Section 2 provides an overview of relevant HLT technologies. Section 3 focuses on automatic metadata extraction and document annotation for the Semantic Web. Section 4 discusses language generation from formal knowledge. Finally, Section 5 argues for the closer integration between infrastructures for HLT and the Semantic Web.

2 The Role of HLT

The web revolution has been based largely on human language materials, and in making the shift to the next generation knowledge-based web, human language will remain key. Human Language Technology involves the analysis, mining and production of natural language. HLT has matured over the last decade to a point at which robust and scaleable applications are possible in a variety of areas, and new projects in the Semantic Web area (e.g. SEKT – `http://sekt.semanticweb.org`) are now poised to exploit this development.

Figure 1 illustrates the way in which Human Language Technology can be used to bring together the natural language upon which the current web is mainly based and the formal knowledge at the basis of next generation Semantic Web.

Information Extraction (IE) is a process which takes unseen texts as input and produces fixed-format, unambiguous data as output. This data may be used directly for display to users, or may be stored in a database or spreadsheet for later analysis, or may be used for indexing purposes in Information Retrieval (IR) applications. It is instructive to compare IE and IR: whereas IR simply finds texts and presents them to the user, the typical IE application analyses texts and presents only the specific information from them that the user is interested in. For example, a user of an IR system wanting information on the share price movements of companies with holdings in Bolivian raw materials would typically

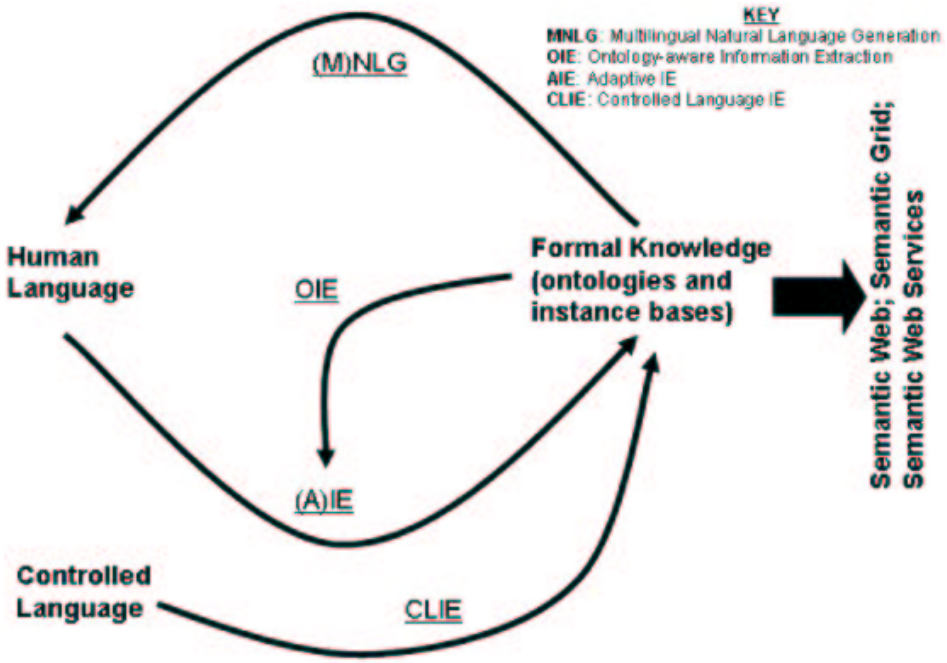


Fig. 1. Closing the language loop

type in a list of relevant words and receive in return a set of documents (e.g. newspaper articles) which contain likely matches. The user would then read the documents and extract the requisite information themselves. They might then enter the information in a spreadsheet and produce a chart for a report or presentation. In contrast, an IE system user could, with a properly configured application, automatically populate their spreadsheet directly with the names of companies and the price movements. The new challenge for IE is to populate ontologies and generate metadata.

Natural Language Generation (NLG) is the inverse of IE: from structured data in a knowledge base NLG techniques produce natural language text, tailored to the presentational context and the target reader[2]. NLG techniques use and build models of the context and the user and use them to select appropriate presentation strategies. For example, deliver short summaries to the user's WAP phone or a longer multimodal text if the user is using their desktop. Similarly, NLG techniques can use simpler terminology and explain unknown terms to the naive user, while different terminology and text style is used for the expert user. The new challenge for NLG is to generate texts from ontologies and metadata, which requires the development of new NLG methods allowing easy portability between domains, based on machine learning.

[2] For an introduction to NLG see [12].

3 From Language to Knowledge

3.1 Ontology-aware Information Extraction

Recently there has been work on using Information Extraction (IE) to help users annotate (semi-)automatically Web pages with semantic content e.g., [10, 8]. The user trains the IE tools by annotating manually some pages, until the system can start suggesting annotations automatically. Then the user can continue to train the system by correcting its errors and/or annotating missed information. These annotation tools however do not provide the user with a way to customise the integrated language technology directly. While many users would not need or want such customisation facilities, users who already have ontologies with rich instance data will benefit if they can make this data available to the IE components.

The more serious problem however, as discussed in [8], is that there is often a gap between the annotations and their types produced by IE and the classes and properties in the user's ontology. The proposed solution is to write some kind of rules, such as logical rules, to achieve this. For example, an IE system would typically annotate London and UK as locations, but extra rules are needed to specify that there is a containment relationship between the two (for other examples see [8]). However, rule writing of the proposed kind is too difficult for most users and a new solution is needed to bridge this gap.

Therefore, the outstanding challenge is to develop tools to provide the user with a way to customise the integrated language technology directly by connecting the IE components to their ontology to make the tools sensitive to future changes in the model and to bridge the gap between IE results and ontology classes. This ontology-aware IE can be configured to provide a service that will annotate any page relative to a particular ontology, so that software agents can use IE services to find instances of concepts from their own models. This removes some need to map between ontologies: the annotator extracts directly to the user's own ontology. The work will need to go beyond state-of-the-art by:

1. Developing support for learning with unlabeled data, adopting recent techniques from within Data Mining, to extract maximum information from the minimal manual input.
2. Developing hybrid adaptive IE tools, combining rule-based and machine learning approaches and using reasoning services, to perform entity tracking within and across documents.

3.2 Controlled Language IE (CLIE)

Creating formal data is a high initial barrier to entry for small organisations and individuals wishing to make data available to semantic knowledge technology. Part of the answer is in authoring tools, but it is also possible that the definition of a controlled language for formal data description will lower this barrier significantly. Building on controlled language MT work, IE for controlled language analysis could achieve the high levels of accuracy necessary to make this viable.

3.3 Semantic Reference Disambiguation

IE systems currently recognise particular entities and relations, but do not resolve them with respect to a given ontology of classes and instances as needed for the Semantic Web. For instance, they recognise Cambridge as an entity of type Location or City, but do not disambiguate it with respect to which real-world entity it is, i.e., Cambridge in the UK or the US or some other new instance not present in the ontology.

Therefore, existing coreference methods need to be extended with new algorithms for semantic reference disambiguation. A variety of techniques can be explored here. First, vector-space models can be used to detect whether the entity in the text occurs in the same context as an instance in the ontology, as has been done in work on cross-document coreference [2]. Another approach could be to apply work on communities of practice from knowledge management [1] and treat the problem as ensuring referential integrity of ontologies. A useful baseline approach is to disambiguate to the most frequent instance as determined by a reference corpus.

3.4 Quantitative Evaluation: Data, Tools and Metrics

An integral part of the development of machine learning approaches for IE is the ability to perform automatic quantitative evaluation in order to measure differences between different versions of the system and also allow comparative evaluation with other approaches. Automatic quantitative evaluation of IE for the Semantic Web requires: an annotated corpus, an evaluation metric and a scoring tool implementing this metric. Existing corpora and evaluation metrics for IE (e.g., those created for the Message Understanding Conferences [13]) are not suitable for evaluating IE tools in the Semantic Web context, because these corpora and metrics only detect very coarse-grained types of entities, without a specific ontology, and without creating a reference between the entities and events in the documents and those that occur in the target ontology.

The challenge is to create corpora and metrics suitable for evaluating the performance of the IE tools specifically on annotating content relative to ontologies. This will include evaluation along several dimensions:

- Detection of entities and events, given a target ontology of the domain.
- Disambiguation of the entities and events from the documents with respect to instances in the given ontology. For example, measuring whether the IE correctly disambiguated "Cambridge" in the text to the correct instance: Cambridge, UK vs Cambridge, MA.
- Decision when a new instance needs to be added to the ontology, because the text contains a new instance, that does not already exist in the ontology.

In order to achieve this, an evaluation corpus, annotated with the correct ontological class and instance, is needed. The corpus needs to consist of two parts – testing and evaluation part, so that the testing part can be used for

system development and testing, while the evaluation one will be used as a gold-standard for evaluation only.

In addition, new metrics for scoring are needed, in order to take into account the nature of the task: for example, the use of ontologies means that correctness is more of a scalar issue, rather than a binary one. The scoring tool needs to automatically compare the system results with the human-annotated standard and produce quantitative measures. In addition, there needs to be a regression testing tool that enables tracking of the system's performance over time, which takes into account relations and distances in the ontology.

4 From Knowledge to Language

NLG can be applied to provide automated documentation of ontologies and knowledge bases. Unlike human-written texts, an automatic approach will constantly keep the documentation up-to-date which is vitally important where knowledge is dynamic and is updated frequently. The NLG tools will also allow generation in multiple languages without the need for human or automatic translation.

The main challenge posed for NLG by the Semantic Web is to provide tools and techniques that are extendable and maintainable (the majority of existing NLG applications can only be modified and extended by specialists). The most promising avenue seems to be the development of novel approaches that combine machine learning with advanced interactive tools for non-specialist users, in order to enhance the adaptivity of NLG.

In addition, the NLG tools can provide context aware and personalised profile-sensitive delivery using state-of-the-art methods for generation of personalised presentations, based on the automatically built user profiles [5]. These methods can effectively summarise knowledge at the appropriate level of granularity and present it in natural language.

Finally, the quantitative evaluation of some NLG methods also poses a challenge due to the lack of corpora, metrics, and evaluation tools [3].

5 Infrastructures, interoperability, and support

Existing HLT infrastructures, such as GATE [7, 6], while offering powerful capabilities, are oriented towards specialists. However, HLT take-up in other fields, like bioinformatics or knowledge technologies, is dependent on tools that offer targetted support for non-experts to customise language processing facilities for their specific domains and tasks.

In addition, a number of HLT fields, e.g., Information Extraction, can also benefit from tools and resources developed in relation to these other fields. For example, ontologies and reasoning services from the Semantic Web can be used as part of the IE task, in order to produce Semantic Web content that is automatically derived from existing data. Also, unsupervised Machine Learning methods

for Information Extraction need digital library resources such as gazetteers and thesauri as a source of readily available training data. Therefore another challenge is to provide interoperation with these infrastructures and services, which in combination will offer far more than any of them on their own.

Finally, infrastructural support for delivery of language processing technology over the Grid and with Web services is needed, in order to parallelise slow operations and to enable embedding of HLT in diverse Semantic Web applications.

The first steps towards providing interoperability between Semantic Web and HLT infrastructures have been carried out as part of the open-source GATE HLT infrastructure [4]. GATE has been extended recently to provide support for importing, accessing and visualising ontologies as a new type of resource available to language processing applications, such as IE. Much of this functionality is provided through the integration of the Protégé editor [11] within the GATE visual environment. Ontology import/export is provided from/to DAML+OIL and the formats supported by Protégé . In addition, the results of any IE application can be exported for the Semantic Web in DAML+OIL format.

Another recent effort in this area is KIM – a Knowledge and Information Management platform [4]. KIM offers an RDF(S) repository for storage and management of both language and Semantic Web data, reasoning services, ontology editing and browsing, semantic query interface, and a browser plug-in for document viewing/annotation.

However, a number of open issues are yet to be solved in this area, the most important of which are helping non-expert users to customise the language technology embedded in their applications and the delivery of HLT as Semantic Web services.

6 Conclusion

This position paper motivated the need for Semantic Web enabled Human Language Technology tools and discussed the major outstanding challenges in this area. It introduced the idea of a "language loop" and showed how HLT can be used to bridge the gap between the current web of language and the Semantic Web. We also argued for a closer integration between HLT and Semantic Web tools and infrastructures.

Progress in the development of the Information Society has seen a truly revolutionary decade. Dot com crash notwithstanding, all our lives have been radically changed by the advent of widespread public networking. We believe that a new social revolution is imminent, involving the transition from Information Society to Knowledge Society. SEKT aims to contribute to this revolution, and to embed language technology at its heart.

References

1. H. Alani, S. Dasmahapatra, N. Gibbins, H. Glaser, S. Harris, Y. Kalfoglou, K. O'Hara, and N. Shadbolt. Managing Reference: Ensuring Referential Integrity of

Ontologies for the Semantic Web. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 317–334, Siguenza, Spain, 2002.

2. A. Bagga and A. W. Biermann. A methodology for cross-document coreference. In *Proceedings of the Fifth Joint Conference on Information Sciences (JCIS 2000)*, pages 207–210, 2000.
3. K. Bontcheva. Reuse and problems in the evaluation of nlg systems. In *Proceedings of EACL03 Workshop on Evaluation Initiatives*, Budapest, Hungary, 2003.
4. K. Bontcheva, A. Kiryakov, H. Cunningham, B. Popov, and M. Dimitrov. Semantic web enabled, open source language technology. In *EACL workshop on Language Technology and the Semantic Web: NLP and XML*, Budapest, Hungary, 2003.
5. Kalina Bontcheva. Tailoring the Content of Dynamically Generated Explanations. In M. Bauer, P. Gmytrasiewicz, and J. Vassileva, editors, *User Modelling 2001*, volume 2109 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, Berling Heidelberg, 2001.
6. H. Cunningham. GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36:223–254, 2002.
7. H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.
8. S. Handschuh, S. Staab, and F. Ciravegna. S-CREAM — Semi-automatic CREAtion of Metadata. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 358–372, Siguenza, Spain, 2002.
9. M. Klein, D. Fensel, A. Kiryakov, and D. Ognyanov. Ontology Versioning and Change Detection on the Web. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 197–212, Siguenza, Spain, 2002.
10. E. Motta, M. Vargas-Vera, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 379–391, Siguenza, Spain, 2002.
11. N.F. Noy, M. Sintek, S. Decker, M. Crubzy, R.W. Fergerson, and M.A. Musen. Creating Semantic Web Contents with Protégé-2000. *IEEE Intelligent Systems*, 16(2):60–71, 2001.
12. E. Reiter and R. Dale. Building Natural Language Generation Systems. *Journal of Natural Language Engineering*, Vol. 3 Part 1, 1999.
13. SAIC. Proceedings of the Seventh Message Understanding Conference (MUC-7). `http://www.itl.nist.gov/iaui/894.02/related_projects/muc/index.html`, 1998.

Multi-strategy Definition of Annotation Services in Melita

Fabio Ciravegna, Alexiei Dingli, Jose‘ Iria, and Yorick Wilks

Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello Street, Sheffield S1 4DP
{fabio|alexiei|jiria|yorick}@dcs.shef.ac.uk
http://www.nlp.shef.ac.uk

Abstract. The definition of methodologies for automatic ontology-based document annotation is a fundamental step in the Semantic Web vision. In the near future, semantic annotation services could become as important as search engines are today. Tools for the easy and effective development of such services are therefore needed. In this paper, we present Melita, a tool for the definition and development of ontology-based annotation services. Melita goes beyond the dichotomy rule learning Vs rule writing of classic annotation systems, as it allows adopting different strategies, from annotating examples in a corpus for training a learner to rule writing and even a mixture of them. It also supports users in defining and maintaining an ontology for annotation and in delivering the annotation service. The result is a tool easy to use and flexible to different user needs.

1 Introduction

The Semantic Web needs semantically-based document annotation to both enable better document retrieval and empower semantically-aware agents. Different annotation schemas are likely to be superimposed on a web page using different ontologies, reflecting different domains of interest over the same information as regarded by different actors. Most of the annotations are likely to be imposed by web actors other than the pages' owner, exactly like nowadays' search engines produce indexes without modifying the page code. In currently available technology, however, annotation is meant mainly to be statically associated to the document. Moreover, most of the available technology is based on human centered annotation, very often completely manual [9]. Manual annotation is difficult, time consuming and expensive [5]. Convincing users to annotate documents for the Web (e.g. using ontologies) is difficult and requires a worldwide action of uncertain outcome.

Static and manual annotation associated to a document is prone to:

- become obsolete, i.e. not aligned with pages' updates or evolving ontologies;
- be incomplete or incorrect with respect to a specific ontology, especially if the human annotator is not skilled enough;

– be irrelevant from the point of view of a specific ontology, e.g. a page in a pet shop web site may be annotated with shop-related annotations, but some users would rather prefer to find annotations related to animals.

Producing methodologies for automatic annotation of pages becomes therefore important. The initial annotation associated to the document (and any other static one) loses its importance because at any time it is possible to automatically (re)annotate the document. In the future Semantic Web, automatic Semantic Annotation Systems (SAS) are likely to become as important as indexing systems are for search engines nowadays.

Automatic annotation methodologies have been developed in the past in different research areas such as Information Extraction from text (IE)[15], wrapper induction [11] and machine learning [14]. Methodologies have been defined for:

– reducing the burden in some SW annotation tools using adaptive IE[16] [8] [5]
– crawling the Web in an unsupervised way for harvesting domain specific information [12] [14][4]
– produce generic annotation such as Named Entity Recognition [7].

These methodologies *per se* represent a partial solution to the problem. Annotation tools based on adaptive IE [16] [8] [5] require the manual (or semi-automatic) annotation of examples to train an underlying adaptive IE system. When the user thinks the IE engine has reached a satisfying accuracy, the annotation service can be released. These tools focus on sequential annotation of documents in a corpus. Goal of the annotation is to train the IE system, which in turn will generate resources (e.g. rules) for the annotation service. They are designed with naive users in mind, i.e. users not acquainted with IE: they allow to produce a SAS via document annotation only. Although many users are able to write satisfying pattern matching rules for many tasks, such tools do not provide any facilities for rule writing. Unfortunately, in some cases a number of texts need to be annotated until the system learns patterns users could summarize in a couple of minutes. In case of data sparseness (e.g. a type of information is rarely present in the texts), users must browse/annotate many documents before a number of relevant cases can be retrieved that is sufficient to produce a reliably trained system also for the sparse phenomena.

Fully or largely unsupervised systems (e.g. [14][4]) exploit the redundancy and/or regularity of corpora (or the web) to automatically produce annotations, using user feedback to retrain the system. User's contribution is generally limited to preliminary lists of names of relevant objects and the correction of final or intermediate results. From the point of view of the generated annotation service, they tend to work as a black box. Users are not generally expected to contribute in the development, for example by writing rules. The problem of data sparseness mentioned above does not apply because the approach is largely unsupervised. Nevertheless the opacity of the rule learner and the inability to operate on it can sometimes be irritating for some users [10].

Systems requiring manual development of rules (e.g. [7]) rely on the user's ability to generalize over examples and to capture regularities in the corpus. The current state of the art in IE shows that the average manual system outperforms machine learning based systems trained on the same amount of data [1]. This is mainly because humans can generalize better and more quickly than systems, i.e. they require less examples to produce a good annotation service. For example Josen et al [10] show how a human inspecting 50 examples of deeds of conveyance was able to slightly outperform an adaptive IE system trained on 200. When using a manual system, however, personnel trained in developing grammars are needed. It is not possible to enable naive users to develop applications and there is no way to learn from any pre-annotated resource that should be available. Moreover a corpus needs in any case to be manually annotated in order to test for the system accuracy. Such corpus is generally of more limited size than the one needed to train an automatic system.

In this paper, we describe a tool integrating two of the approaches mentioned above (rule writing and document annotation) in an integrated way. Users can either write patterns or annotate texts and run a learner (or both) according to their skills and momentary needs. Moreover, the tool supports users in a number of other steps in the definition of a SAS: from the definition/refinement of the ontology, to (assisted) document annotation, to the writing of annotation patterns to the delivery of annotation service.

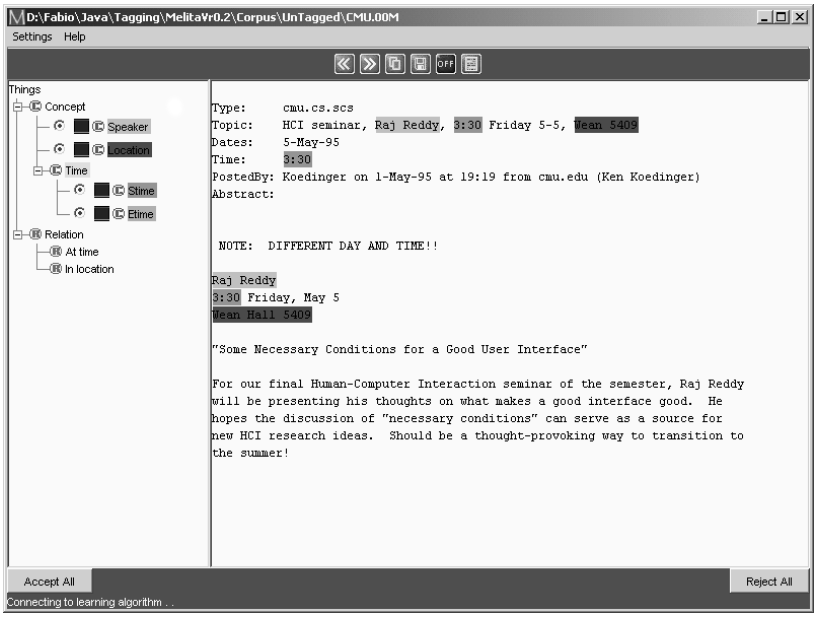


Fig. 1. The Melita Interface in the previous version: on the left the ontology is shown. On the right the document to be annotated is available. To annotate users select a concept and then use the mouse to highlight the relevant part of the document.

2 Melita

The current work extends Melita [5], an ontology-based text annotation tool. Melita's main control panel is depicted in figure Figure 1. It is composed of two main areas:

- the ontology (left) representing the annotations that can be inserted; annotations are associated to concepts and relations. A specific color is associated to each node in the ontology (e.g. in Figure 1 the concept "speaker" is depicted in gray);
- the document to be annotated (center-right). Selecting the the node in the ontology and then clicking on portion of text with the mouse inserts annotations. Inserted annotations are shown by turning the background of the annotated text portion to the color associated to the node in the hierarchy (e.g. the background of the portion of text representing a speaker becomes gray).

Melita provides support to annotation based on adaptive Information Extraction (IE). While the user annotates, an IE system (Amilcare [3]) monitors the annotations inserted by the user and - when similar cases are found in new documents - suggests annotations to the user.

The goal of Melita is to provide a way to produce annotation services using only knowledge of the domain. Melita was originally designed with naive users in mind. It does just require the annotation of documents using an intuitive interface, but it does not support users in an integrated way - it does not enable users to the manage IE rules or the ontology themselves, for instance. Melita supports annotation based on documents. Each document is annotated separately. There is no concept of corpus, apart from the set of documents already annotated that are used to train Amilcare. It is possible to browse documents from a corpus, but it is never possible to query the corpus in its entirety. In the next section we will describe how Melita was enhanced in order to overcome the limitations just mentioned.

3 The three focuses of interaction

A new version of Melita was designed and implemented that supports different tasks and interaction strategies for producing a SAS, from the definition/refinement of the annotation ontology, to (assisted) document annotation, to the writing of annotation patterns, to the delivery of the annotation service.

There are three focuses of interaction for the user:

- the ontology;
- the corpus, both as a whole and as a collection of single documents;
- the annotation pattern grammar(s), either user- or system-defined.

Users can move the focus and the methodology of interaction during the creation of the SAS in a seamless way, for example moving from a focus on document annotation, to rule writing, to ontology editing.

3.1 Initial Setup

Two objects are needed to start the definition of a new SAS in Melita: a corpus and an initial ontology. The ontology defines the labels to annotate documents in the corpus. The corpus will provide the material to train the underlying IE system or to help users in developing rules. The initial ontology can either be provided by the user or learned using (semi-)automatic methodologies [13] [2]. Melita can read ontologies in DAML+OIL, RDF and XI. In the average application this ontology generally represents an initial draft to be refined after exploring some of the documents in the corpus to be annotated.

3.2 Corpus Focus

The corpus provides the material to train the underlying IE system or to help users in developing rules. It also often motivates the refinement of the ontology. Melita provides facilities to access the corpus in three modes:

- browsing/exploring the set of documents;
- annotating single documents; documents are accessed in a sequential mode and annotated using labels from the ontology. The inserted annotations are passed to the learner to induce patterns; these will constitute the backbone of the first version of the annotation service. This modality is the classic provided by many annotation tools such as MnM [16], Ontomat [8] and the previous version of Melita.
- querying, to retrieve documents containing interesting information or exploring potential patterns in the corpus;

Here the focus is no longer only on the sequence of documents to annotate, but also the whole corpus can be queried and browsed as a whole. This kind of facilities is not included in any of the annotation tools we have seen so far. We will come back to querying in Section 3.4.

3.3 Ontology Focus

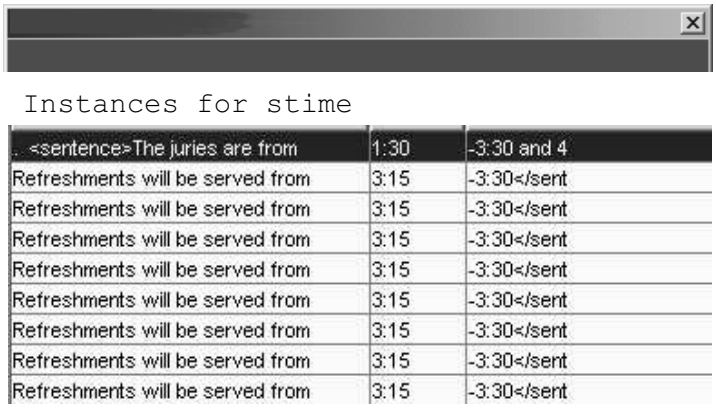
Focussing on the ontology is important to get a global view of the status of the provided annotation, for example by inspecting all the instances associated to a specific concept/relation that have been annotated in the corpus. This may be useful to:

- revise the ontology, e.g. decide to introduce a new concept or to split an existing one into two subconcepts;
- check the kind of phenomena or instances discovered so far in the corpus.
- understand the level of coverage of the current patterns, both induced and written (see next subsection);

During annotation, and especially at the beginning of the process, users may need to evolve the ontology. Users can for example realize that a specific concept should be actually split in two different subconcepts. Melita enables the user:

– to decide about changes to the ontology in an informed way: users can inspect all the instances of/annotation for a specific concept as they have been identified so far (or even inspecting some new ones retrieved querying the corpus), analyze the documents in which they appear ("the context of the information") and decide for or against the possible modifications to the ontology.
– to rearrange the inserted annotations to the changes in the ontology in an efficient and effective way; for example if a concept is split all the annotations for that concept should be presented and the user should be enabled to change them according to the concept definition. The same applies to the patterns the users should have provided so far.

The focus on the ontology implies moving away again from the focus on single documents. The focus on the instances as annotated in the corpus as a whole (as opposed to annotated in a single document); see figure 2. Users are enabled to change the annotations without flipping through all the documents, but just focussing either on the instances themselves or to inspect just the portions of documents involved (not the whole documents) and eventually change the annotation associated to those instances.


Instances for stime

. <sentence>The juries are from	1:30	-3:30 and 4
Refreshments will be served from	3:15	-3:30</sent
Refreshments will be served from	3:15	-3:30</sent
Refreshments will be served from	3:15	-3:30</sent
Refreshments will be served from	3:15	-3:30</sent
Refreshments will be served from	3:15	-3:30</sent
Refreshments will be served from	3:15	-3:30</sent
Refreshments will be served from	3:15	-3:30</sent
Refreshments will be served from	3:15	-3:30</sent

Fig. 2. The list of instances associated to a concept. They are represented as they appear in the corpus (each line is an instance found in the corpus). It is possible to inspect the whole document in which the instance was found by clicking on the instance.

3.4 Grammar Focus

The grammar is the real goal of development in generating a SAS, being the resource enabling the final service. As mentioned, in the classic approach the IE system works as a black box, i.e. no rule modification is possible. Many users are keen to develop rules when they feel that this allows converging more rapidly to an optimal annotation service. This is because the average IE system may need to

see more examples than a person before converging to optimal rules. For example we noted that for recognizing a specific time expression the old Melita required the annotation of 2 or 3 dozens of examples in order to obtain 75% recall and 95% precision [6]. An average user will be able to derive very efficient patterns with a handful of examples, simply using their common sense knowledge about time expressions. Enabling users to write patterns (or to modify the induced ones) when they can/want can drastically reduce the time for producing the annotation service.

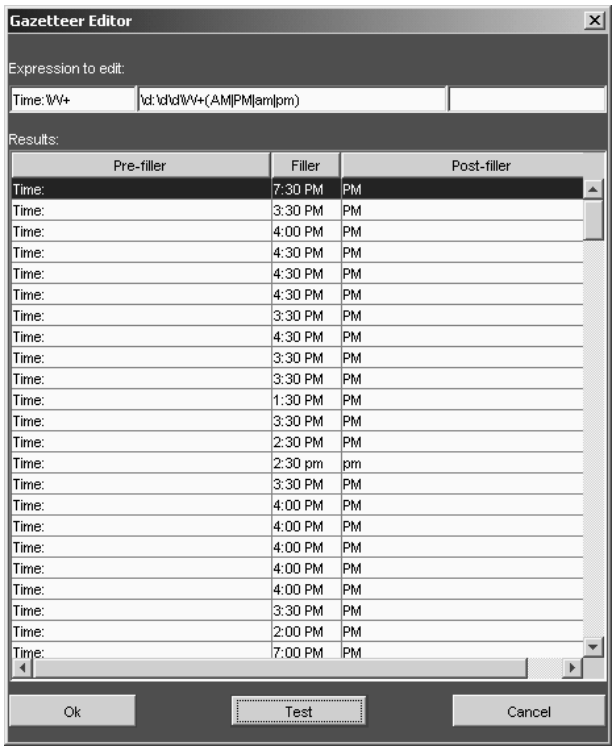


Fig. 3. The editor for regular expression showing matches for the patterns. A pattern is composed of a filler (center) and the contexts (left/right).

In Melita there are two types of patterns users can develop:

- classification patterns aimed at retrieving other pieces of information to reason on, either during IE pattern writing or during ontology revision; this feature is also useful in case of data sparseness in order to retrieve further documents to be annotated. These patterns tend to be generic patterns able to identify likely interesting information; they are not expected to be particularly accurate or to become as-is part of the annotation service.
- IE patterns aimed at incrementing the capabilities of the resulting annotation service, i.e. proper extraction patterns the user is contributing to the

IE learner. When a pattern is accepted, all the examples covered are automatically annotated in the corpus. IE patterns can either be compiled from scratch or be modified versions of induced patterns.

Accordingly, two types of editors are provided in Melita:

- a regular expression editor matching strings (mainly thought to be used for querying but that can be used for extraction patterns as well); this is an editor for users not particularly acquainted with Natural Language Processing (see Figure 3);
- an editor for patterns accessing the NLP features used by Amilcare; these features are derived by a linguistic preprocessor, e.g. Part of Speech Tagging, Gazetteer information and Named Entity Recognition, etc. The Amilcare's preprocessor is based on Annie [7]).

Pattern writing requires to focus on the corpus as a whole: patterns need to be tested on the whole corpus (as opposed to single documents) to check their effectiveness. Users need to identify positive and negative examples covered by the patterns, either from the annotated documents or from the non-annotated ones. Facilities for testing patterns and presenting results are then needed. Such facilities are not present in any of the document-based annotation tools mentioned above. This is a modality that is typical of the tools that require rule writing.

3.5 Closing the Loop

The different views mentioned above tend to be quite separated in existing systems. For example in both the previous version of Melita and MnM [16] most of the functionalities mentioned above for document annotation are provided, but no facilities for modifying the ontology exist (for which other tools must be used) or to cope with the corpus as a whole (querying can be obtained using a search engine), and no rules can be edited (but Amilcare's rule manager and editor can be used). The use of different tools makes very difficult switching among the different modalities and focuses.

In the new Melita, it is possible to move from the different views in a seamless way. When in corpus view, it is possible to access instances in the corpus and therefore moving to the view on the ontology. From the instances in the corpus is also possible to move to the patterns that recognize them (if any) and therefore moving to the focus on the grammar. When in grammar view, it is possible to move to the corpus view via the annotations inserted by the patterns in the corpus. Using the recognized instances it is possible to move to the ontology view. When in ontology view it is possible to inspect all the instances and how they are presented in the corpus (and therefore allowing to move on the corpus view) or the patterns used to recognize them (and therefore moving to the grammar view).

4 Conclusion and Future Work

Melita is a tool for defining and developing automatic ontology-based annotation services that provides different views over the task, based on three perspectives: the corpus to be used to develop the annotation service, the reference ontology and the patterns and grammars for annotation. In summary the following facilities are provided:

- Corpus:
  - sequential document annotation;
  - corpus querying (using patterns from the grammar);
  - from annotations to instances in the ontology (move to ontology view);
  - from annotation to the rules that inserted them (move to grammar view);
- Ontology:
  - ontology loading and editing (additions, modifications, etc.);
  - inspection of all the rules associated to an instance (move to grammar view);
  - inspection of how a (set of) instance(s) is presented in the corpus (move to corpus view);
- Grammar
  - grammar management (adding and removing rules)
  - rule editing;
  - rule testing and debugging;
  - inspection of all the instances associated to a rule (move to Corpus or Ontology view);

Melita allows to exploit the user abilities at their best. IE experts will mainly focus on developing rules, while non-IE experts will mainly annotate texts. Ontology engineers will use it to help validate the ontology. Average users will probably use a mixed strategy. We are organizing experiments to classify the behavior of different users and to quantify the gain provided by the different views with respect to the previous version. Details on experiments using the previous version can be found in [5] and [6].

Future developments will concern the inclusion of facilities from the Armadillo tool [4]. Armadillo is a system for unsupervised information extraction and integration from large repositories. Armadillo could be used to retrieve new documents from external sources (e.g. the Web) and apply/check the existing patterns. This could be useful for reducing the impact of data sparseness due to limitations in corpus size [4].

Melita is currently used by a UK company to generate an anonymization service for hospital patient records. They final service will discover, annotate and anonymize both the patient's personal details and specific events that could allow identifying the patient. The cleaned records will then be made available for research in medicine. Melita uses as underlying IE system Amilcare [3], an adaptive IE tool specifically designed for document annotation that has been integrated also in MnM and SCREAM and it is currently under use in a dozen of industrial and academic sites.

Acknowledgements

This work is partially funded by AKT project (http://www.aktors.org), sponsored by the UK Engineering and Physical Sciences Research Council (grant GR/N15764/01). AKT involves the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University. Its objectives are to develop advanced technologies for knowledge management.

The work is also partially funded by the IST-dot.kom project (http://www.dot-kom.org), sponsored by the European Commission as part of the framework V, (grant IST-2001-34038). dot.kom involves the University of Sheffield (UK), ITC-Irst (I), Ontoprise (D), the Open University (UK), Quinary (I) and the University of Karlsruhe (D). Its objectives are to develop Knowledge Management and Semantic Web methodologies based on Adaptive Information Extraction from Text.

References

1. Douglas E. Appelt and David Israel. Introduction to information extraction technology. IJCAI-99 Tutorial, August 2 1999. Stockholm, Sweden, "http://www.w3.org/TR/1998/REC-xml-19980210".
2. Christopher Brewster, Fabio Ciravegna, and Yorick Wilks. User-centred ontology learning for knowledge management. In *Proceedings of the 7th International Conference on Applications of Natural Language to Information Systems*, Stockholm, June 2001. Lecture Notes in Computer Sciences, Springer Verlag.
3. Fabio Ciravegna. Designing adaptive information extraction for the semantic web in amilcare. In S. Handschuh and S. Staab, editors, *Annotation for the Semantic Web*, Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam, 2003.
4. Fabio Ciravegna, Alexiei Dingli, David Guthrie, and Yorick Wilks. Integrating information to bootstrap information extraction from web sites. In *Proceedings of the IJCAI 2003 Workshop on Information Integration on the Web, workshop in conjunction with the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003)*, 2003. Acapulco, Mexico, August, 9-15.
5. Fabio Ciravegna, Alexiei Dingli, Daniela Petrelli, and Yorick Wilks. User-system cooperation in document annotation based on information extraction. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02*. Springer Verlag, 2002.
6. Fabio Ciravegna, Alexiei Dingli, Yorick Wilks, and Daniela Petrelli. Using adaptive information extraction for effective human-centred document annotation. In I.Renz J.Franke, G.Nakhaeizadeh, editor, *Text Mining, Theoretical Aspects and Applications*, Lecture Notes in Computer Science. Springer Verlag, 2003.
7. H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan". Gate: an architecture for development of robust hlt". In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)"*, July" 2002.
8. S. Handschuh, S. Staab, and F. Ciravegna. S-CREAM - Semi-automatic CREAtion of Metadata. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02*. Springer Verlag, 2002.

9. S. Handschuh, S. Staab, and A. Maedche. CREAM — Creating relational metadata with a component-based, ontology driven framework. In *In Proceedings of K-Cap 2001*, Victoria, BC, Canada, October 2001.
10. M. Josen, P. Jongejan, G.J. Kersten, and E. Zopfi. Towards complexity measures: Guidelines for the feasibility of information extraction projects. In *Proceedings of the Dutch Conference on Artificial Intelligence*, 2001.
11. N. Kushmerick, D. Weld, and R. Doorenbos. Wrapper induction for information extraction. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1997.*, 1997.
12. Thomas Leonard and Hugh Glaser. Large scale acquisition and maintenance from the web without source access. In Siegfried Handschuh, Rose Dieng-Kuntz, and Steffen Staab, editors, *Proceedings Workshop 4, Knowledge Markup and Semantic Annotation, K-CAP 2001*, 2001.
13. A. Maedche and S. Staab. Semi-automatic engineering of ontologies from text. In *Proceedings of the 12th Internal Conference on Software and Knowledge Engineering. Chicago, USA, July, 5-7, 2000.* KSI, 2000.
14. Tom Mitchell. Extracting targeted data from the web. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, San Francisco, California, 2001.
15. Maria Teresa Pazienza, editor. *Information Extraction: A multidisciplinary approach to an emerging information technology.* Springer Verlag, 1999.
16. M. Vargas-Vera, Enrico Motta, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna. MnM: Ontology driven semi-automatic or automatic support for semantic markup. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02.* Springer Verlag, 2002.

Talking OWLs: Towards an Ontology Verbalizer

Graham Wilcock

University of Helsinki
00014 Helsinki, Finland
graham.wilcock@helsinki.fi

Abstract. The paper describes on-going work on an ontology verbalizer which can provide spoken summaries and explanations of the information specified in ontologies. The approach combines semantic web techniques with natural language generation and text-to-speech.

1 Previous work on generation from ontologies

Referring to ontologies formalized in Ontolingua, Aguado *et al.* say:

> Our experience shows that domain experts and human final users do not understand formal ontologies codified in such languages even if such languages have a browser and a graphic user interface to display the ontology content. [1]

They describe a system that translates the ontology into natural language to help users understand it. To map from domain concepts to linguistic representations they use the Generalized Upper Model, based on the Penman Upper Model [2], as a "linguistic ontology". Surface realization is done with KPML.

Frohlich and Riet [4] describe domain independent tools for generation based on "using different ontologies to represent the domain knowledge for different tasks of the generation process." Like [1], they have a domain-specific layer at the top and a domain-independent layer based on the Penman Upper Model at the bottom, with KPML for surface realization.

These earlier projects used languages and tools such as Ontolingua, Penman Upper Model, LISP, LOOM and KPML. Although we can now use Java, XML, RDF and OWL, we still need to help users to understand the ontologies.

2 Generating summaries from RDF

In XML-based natural language generation [9], [10], [11], a pipeline of XSLT transformations implements the sequence of processing stages in an orthodox pipeline architecture for natural language generation. At the start of the pipeline, XSLT template-based generation creates an XML text plan tree whose leaves are domain concept messages. The text plan tree is transformed by the microplanning stages into an XML text specification tree whose leaves are linguistic phrase

specifications. The XSLT processors are embedded in Java, using SAX events to pass XML content efficiently down the pipeline.

XML-based generation has been used in a spoken dialogue system [6]. For spoken output, the final realization stage produces JSML (Java Speech Markup Language) [7] which is XML-based. The JSML is passed to FreeTTS [8], a speech synthesizer implemented entirely in Java.

XML-based generation can naturally be used for generation from RDF. A prototype implementation uses Jena [5] to feed content from RDF into the XSLT pipeline. Jena includes an RDF parser (ARP), an RDF query language (RDQL), and support for persistent storage of RDF models in relational databases.

```
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
         xmlns:vcard='http://www.w3.org/2001/vcard-rdf/3.0#'>
  <rdf:Description rdf:about='http://somewhere/JohnSmith/'>
    <vcard:FN>John Smith</vcard:FN>
    <vcard:N rdf:nodeID='A0'/>
    <vcard:EMAIL rdf:nodeID='A1'/>
  </rdf:Description>
  <rdf:Description rdf:nodeID='A1'>
    <rdf:value>John@somewhere.com</rdf:value>
    <rdf:type rdf:resource='http://www.w3.org/2001/vcard-rdf/3.0#internet'/>
  </rdf:Description>
  <rdf:Description rdf:nodeID='A0'>
    <vcard:Family>Smith</vcard:Family>
    <vcard:Given>John</vcard:Given>
  </rdf:Description>
</rdf:RDF>
```

Fig. 1. Example RDF description from the Jena tutorial

The simple RDF description in Figure 1 is taken from the Jena tutorial [5]. It describes a specific person (John Smith), not the general class of persons, and it uses the RDF encoding for vCard (visiting card) personal information [13]. If the natural language generator were limited to the information given explicitly in the RDF representation, it might produce something like Example 1.

Example 1.
This is a description of 'http://somewhere/JohnSmith/'. The description includes 3 items: 'vcard:FN', 'vcard:N' and 'vcard:EMAIL'.

The value of 'vcard:FN' is 'John Smith'.

The description of 'vcard:N' includes 2 items: 'vcard:Family' and 'vcard:Given'. The value of 'vcard:Family' is 'Smith'. The value of 'vcard:Given' is 'John'.

The description of 'vcard:EMAIL' includes a value and a type. The value is 'John@somewhere.com'. The type is 'http://www.w3.org/2001/vcard-rdf/3.0#internet'.

However, the generator can exploit the use of vCard by providing predefined XSLT text plan templates for vCard, following the domain-specific approach of *shallow generation* [3], [10]. The values from the RDF representation are copied into the slots in the text plan template. By using knowledge about vCard, the generator can create a much better text plan equivalent to Example 2.

Example 2.
This is a description of John Smith identified by 'http://somewhere/JohnSmith/'. John Smith's given name is 'John'. John Smith's family name is 'Smith'. John Smith's email address is 'John@somewhere.com'. John Smith's email address is type 'internet'.

Of course, natural language generation can produce something more natural than this. The referring expressions stage of the generator can convert the text plan into a text specification equivalent to Example 3.

Example 3.
This is a description of John Smith identified by 'http://somewhere/JohnSmith/'. His given name is 'John'. His family name is 'Smith'. His email address is 'John@somewhere.com'. It is 'internet' type.

Further, by performing sentence aggregation, the microplanning stages of the generator can produce a text specification equivalent to Example 4.

Example 4.
This is a description of John Smith identified by 'http://somewhere/JohnSmith/'. His given name is 'John' and his family name is 'Smith'. His email address, which is 'internet' type, is 'John@somewhere.com'.

3 Generating explanations from DAML+OIL

The approach described in Section 2 is a form of shallow generation. One of the ideas in shallow generation [3] is to build domain-specific and task-specific generators, and not to attempt general solutions.

Naturally, shallow generation is compatible with a domain-specific ontology, but at first sight it seems incompatible with more general ontologies. However, Aguado *et al.* [1] claim that their rhetorical schemas represent standard patterns of scientific discourse, and they identified a number of stereotypical paragraph templates including definitions, comparisons, examples and classifications. If a small number of explanation schemas are sufficient to generate explanations from ontologies, then shallow generation can be used. This is an important point, to be investigated further.

The approach used for RDF can again be extended to process DAML+OIL. Jena [5] provides Java methods to read a DAML+OIL ontology and load it as a Jena model. There are also Jena methods to list all the ontology classes and to list all the properties. This provides a starting point for verbalising the ontology

contents, but raw lists of classes and properties are very difficult to understand. In order to generate something which is an *explanation* of the ontology, the classes and properties need to be organised into meaningful groups.

This is on-going work. The RDF examples, the DAML+OIL processing, and the use of ontologies in spoken dialogue systems are discussed further in [12]. The current prototype uses RDF and DAML+OIL with Jena 1. Future work will use RDF and OWL[1] with Jena 2.

References

1. G. Aguado, A. Bañón, J. Bateman, S. Bernardos, M. Fernández, A. Gómez-Pérez, E. Nieto, A. Olalla, R. Plaza, and A. Sánchez. Ontogeneration: Reusing domain and linguistic ontologies for Spanish text generation. In *Proceedings of ECAI-98 Workshop on Applications of Ontologies and Problem-solving Methods*, pages 1–10, Brighton, 1998.
2. J. Bateman, R. Kasper, J. Moore, and R. Whitney. A general organization of knowledge for natural language processing: the PENMAN upper model. Technical report, USC/ISI, 1990.
3. S. Busemann and H. Horacek. A flexible shallow approach to text generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, pages 238–247, Niagara-on-the-Lake, Ontario, 1998.
4. M. Fröhlich and R. van de Riet. Using multiple ontologies in a framework for natural language generation. In *Proceedings of ECAI-98 Workshop on Applications of Ontologies and Problem-solving Methods*, pages 67–77, Brighton, 1998.
5. HP Labs. Jena Semantic Web Toolkit. http://www.hpl.hp.com/semweb/jena.htm, 2003.
6. K. Jokinen and G. Wilcock. Confidence-based adaptivity in response generation for a spoken dialogue system. In *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue*, pages 80–89, Aalborg, Denmark, 2001.
7. Sun Microsystems. Java Speech Markup Language Specification, version 0.6. http://java.sun.com/products/java-media/speech/, 1999.
8. Sun Microsystems. FreeTTS: A speech synthesizer written entirly in the Java programming language. http://freetts.sourceforge.net/, 2002.
9. G. Wilcock. Pipelines, templates and transformations: XML for natural language generation. In *Proceedings of the 1st NLP and XML Workshop*, pages 1–8, Tokyo, 2001.
10. G. Wilcock. XML-based Natural Language Generation. In *Towards the Semantic Web and Web Services: XML Finland 2002 - Slide Presentations*, pages 40–63, Helsinki, 2002.
11. G. Wilcock. Integrating Natural Language Generation with XML Web Technology. In *Proceedings of the Demo Sessions of EACL-2003*, pages 247–250, Budapest, 2003.
12. G. Wilcock and K. Jokinen. Generating responses and explanations from RDF/XML and DAML+OIL. In *Knowledge and Reasoning in Practical Dialogue Systems*, pages 58–63. IJCAI-2003, Acapulco, 2003.
13. World Wide Web Consortium. Representing vCard Objects in RDF/XML. http://www.w3.org/TR/vcard-rdf, 2001.

[1] I thank Lauri Carlson for the phrase "Talking OWLs" in the title.

Combining Data Integration with Natural Language Technology for the Semantic Web

Dean Williams, Alexandra Poulovassilis
{dean,ap}@dcs.bbk.ac.uk

School of Computer Science and Information Systems, Birkbeck College, University of London

1 Introduction

The Semantic Web requires us to be able to integrate information from a variety of sources, including unstructured text from web pages, semi-structured XML data, structured databases, and metadata sources such as ontologies. Integration of heterogeneous data sources is a problem that has been addressed by several recent data integration systems, one of which is the AutoMed system being developed at Birkbeck and Imperial Colleges (http://www.doc.ic.ac.uk/automed). In data integration systems, several data sources, each with an associated local schema, are integrated to form a single virtual database with an associated global schema. If the data sources conform to different data models, then these need to be transformed into a common data model as part of the integration process. The AutoMed system uses a low-level graph-based data model, the HDM, as its common data model, and bi-directional schema transformation pathways to transform and integrate heterogeneous schemas [3].

There is clearly potential for using this approach for information integration in the Semantic Web, but a number of extensions are required. In particular, while data in a wide range of structured and semi-structured formats has been dealt with by previous data integration systems, natural language sources and ontologies have not. In this paper we present a method of extracting data and metadata from natural language sources and integrating it with other structured and semi-structured data sources. We describe how existing metadata can be used to assist in this extraction process.

Our approach combines Information Extraction (IE) technology with the AutoMed data integration system. The resulting system, called **ESTEST (Experimental Software for Extracting Structure from Text)**, makes use of existing metadata such as database schemas, natural language ontologies, and domain-specific ontologies, to assist the IE process from text. The Resource Description Framework (RDF) is an emerging standard for representing and sharing ontolological data, and in [6] we have shown how RDF and RDFS can be represented in the HDM, so that RDF/S description bases can be treated as AutoMed data sources. In ESTEST, once new data and new metadata have been extracted from the text, they are integrated with the existing data and metadata. This extraction and integration process can be reiterated as required.

While the new data and new metadata discovered by ESTEST may be expressed in a variety of data models, these can all be mapped into the HDM, and hence for ESTEST we have developed a native HDM repository to store all the new data and metadata, described in [4].

2 ESTEST

ESTEST makes use of AutoMed for its data integration aspects and of an IE system to extract structured information from text. Figure 1 illustrates the ESTEST system architecture, and below we briefly dscribe each of its main components. We refer the reader to the full paper [5] for more details and for an extended example illustrating the use of ESTEST in the area of Road Traffic Accident analysis.

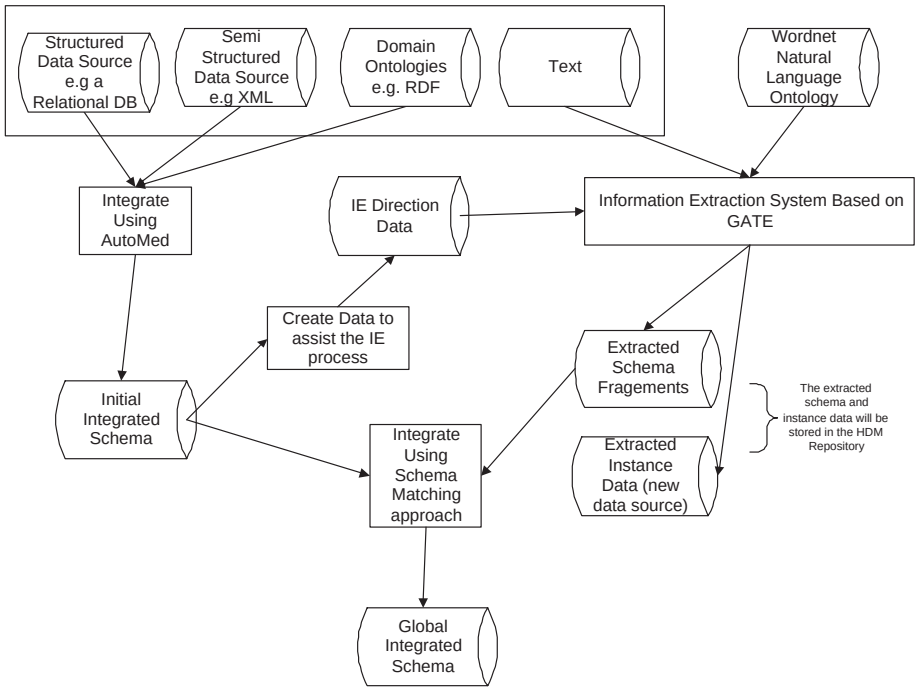


Fig. 1. Overview of the ESTEST System

Initial Integration. The available data sources other than the text (e.g. structured, semi-structured, and domain ontologies) are first integrated into a single virtual global schema, using AutoMed schema transformation pathways. This global schema can then be queried by submitting queries expressed in AutoMed's IQL query language to its global query processor [1].

Create Data to Assist the IE Process. The global virtual resource can be used to provide data which assists the IE process. For example, lists of entities

can be created by submitting queries to the global schema. These lists can then be used by the 'named entity recogniser' components of the IE system (see below). From the global schema, we can also extract information to create templates for grammars. This is described in more detail in the next section, and is based on extracting text information from the global schema e.g. from table and column names in relational schemas.

Information Extraction System Based on GATE. The IE component of ESTEST is based on the GATE system (`http://www.gate.ac.uk`) which allows for a sequence of language processing components to be assembled and marks up annotations on the input text. GATE's language processing components include standard components such as sentence splitters and named entity recognisers. Bespoke components can also be constructed and integrated with the existing standard ones.

We are developing new IE components which will generate templates for the extraction, based on the assumption that the entity types in the existing schema and domain ontologies will be at least a significant subset of the entity types for which we wish to extract information from the text. We are also developing a WordNet component which will make use of its synonym and hyponym structures to allow for alternative lists for words to be found in cases where the textual descriptions of schema elements is restricted, for example to a word in a column name.

The result of the IE process is a set of named annotations over sections of the text. These annotations can be thought of as discovered fragments of schema. These fragments and the text to which they refer are stored in the Extracted Schema Fragments and Extracted Instance Data store, respectively, both of which are implemented using our HDM repository.

Integrate New and Existing Metadata Using Schema Matching. A schema matching algorithm takes each new extracted schema fragment and finds its best match with respect to the global schema, or allows for it to be appended to the global schema. Unlike many other schema matching applications, there will not be much structural information available to assist the matching algorithm and we will rely primarily on element names. However, we will also experiment with using the new instance values extracted, looking to see if these are already present in the extents of candidate schema entities and using any presence to provide evidence of a semantic match.

The Extracted Schema Fragments are integrated with the virtual global schema by means of AutoMed schema transformation pathways which are automatically generated by the above schema matching process. The data in the Extracted Data store can thus be treated as a new AutoMed data source, and queries posed against the virtual global schema will automatically make use of this new data.

3 Conclusion

In this paper we have described the ESTEST system, which extends traditional data integration systems by combining the AutoMed data integration approach with IE technology in order to allow information from ontologies and natural language sources to be integrated with other, semantically related, structured or semi-structured data. ESTEST uses related schema and ontology information to assist the IE process from text. The information extracted is integrated as a new data source with respect to a virtual global schema.

We believe that AutoMed is well-suited to data integration on the Semantic Web:

- The low-level, graph-based nature of the HDM lends itself naturally to modelling both structured and semi-structured information, and for discovering structure in text where both the schema and the instance data may be extended as part of the discovery process.
- AutoMed's bidirectional schema transformation pathways result in easy support of schema evolution, both of local data sources and of integrated virtual schemas (see [2, 3]); this is very likely to be needed in the dynamic environment of Web applications.
- AutoMed's fine-grained schema transformations make transformation pathways amenable to automatic or semi-automatic generation.

We are currently finishing a first implementation of the ESTEST system and will evaluate its effectiveness in a number of application areas, including Road Traffic Accident Data and Operational Intelligence Police Reports. There are a number of research directions for further work, including the use of metadata to drive IE, and schema matching where only text and metadata is available.

References

1. E. Jasper, A. Poulovassilis, and L. Zamboulis. Processing IQL Queries and Migrating Data in the AutoMed toolkit. Technical report, AutoMed Project, 2003.
2. P.J. McBrien and A. Poulovassilis. Schema evolution in heterogeneous database architectures, a schema transformation approach. In *Proc. CAiSE'02, LNCS 2348*, pages 484–499, 2002.
3. P.J. McBrien and A. Poulovassilis. Data integration by bi-directional schema transformation rules. In *Proc. ICDE'03*, 2003.
4. D. Williams. The Automed HDM data store. Technical report, Automed Project, 2003.
5. D. Williams and A.Poulovassilis. Combining data integration with natural language technology for the semantic web. Technical report, Automed Project, 2003.
6. D. Williams and A.Poulovassilis. Representing RDF and RDF Schema in the HDM. Technical report, Automed Project, 2003.

Towards A Language Infrastructure for the Semantic Web

Paul Buitelaar[♦], Thierry Declerck[♦], Nicoletta Calzolari[◊], Alessandro Lenci[*]

[♦]DFKI Language Technology, Stuhlsatzenhausweg 3,
D-66123 Saarbrücken, Germany
{paulb,declerck}@dfki.de

[◊]Istituto di Linguistica Computazionale (ILC) - CNR
Area della Ricerca CNR, Via Alfieri 1 (San Cataldo)
I-56010 PISA, Italy
glottolo@ilc.cnr.it

[*]Dipartimento di Linguistica, Università degli Studi di Pisa
Pisa, Italy
alessandro.lenci@ilc.cnr.it

1 Introduction

In recent years, the Internet evolved from a global medium for information exchange (directed mainly towards human users) into a "global, virtual work environment" (for both human users and machines). Building on the world-wide-web, developments such as *grid technology*, *web services* and the *semantic web* contributed to this transformation, the implications of which are now slowly but clearly being integrated into all areas of the new digital society (e-business, e-government, e-science, etc.) In particular, grid technology allows for distributed computing, web services for a distributed workflow, and the semantic web for increasingly intelligent and therefore autonomous processing.

In this, it is important to realize that the semantic web will function more and more as the man-machine interface of this "global, virtual work environment". The underlying semantic web infrastructure of shared knowledge (ontologies) and markup of resources and services with such knowledge (ontology-based metadata) ensures that a common understanding will exist between the human user and the machine-based processes. However, as much

of human knowledge is and will be encoded in language, multilingual and multicultural aspects (culture as specific to countries, regions and nations, connected with language) will play an important role in establishing and maintaining such common understanding. Given these considerations, we emphasize the following two important issues in future semantic web development:

- **Making the semantic web accessible in many languages:** Authoring support for automatic knowledge markup should be available for many languages thereby avoiding that only documents in some languages will become part of the semantic web

- **Allowing the semantic web to represent many different cultures:** Ontologies should express concepts as used in different cultures, thereby avoiding that the semantic web would force an unnecessary semantic standardization. Therefore, tools for ontology adaptation and for mapping different ontologies should be an integral part of the semantic web infrastructure.

In both cases, there will be an important role for a combination of language technology, ontology engineering and machine learning, in order to provide text analysis for knowledge markup and text mining facilities for ontology mapping and learning. A growing integration of language technology tools into semantic web applications is therefore to be expected with the following characteristics:

- **Language Technology for the Semantic Web:** Language technology tools will be used for efficient, (semi-)automatic knowledge markup (based on information extraction) and ontology development (based on text mining), allowing web documents in many languages and from different cultural backgrounds to be integrated on a large scale within the semantic web.

- **The Semantic Web for Language Technology:** Semantic web methodologies (metadata, web services) and standards (RDF/S, OWL) will be used in the specification of web-based, standardized language resources – data (corpora, lexicons, grammars) and tools – allowing for a

distributed and widespread use of these resources in semantic web applications.

2 Language Technology for the Semantic Web

As human language is a primary mode of knowledge transfer, a growing integration of language technology tools into semantic web applications is to be expected. Language technology tools will be essential in scaling up the semantic web by providing automatic knowledge markup support (e.g. Amilcare, GATE, OntoMat, Melita, MnM) and facilities for ontology monitoring and adaptation (e.g. TextToOnto, OntoLearn, OntoLT). Obviously, it will then be of political and cultural importance that such authoring support for automatic knowledge markup will be available for many languages, thereby avoiding that only documents in some languages will become part of the semantic web.

Ontologies, as used in knowledge markup, are views of the world that tend to evolve rapidly over time and between different applications. Currently, ontologies are often developed in a specific context with a specific goal in mind. However, it is ineffective and costly to build ontologies for each new purpose each time from scratch, which may cause a major barrier for their large-scale use in knowledge markup for the Semantic Web. Creating ambitious semantic web applications based on ontological knowledge implies the development of new, highly adaptive and distributed ways of handling and using knowledge that enable existing ontologies to be adaptable to new environments. Besides time and place this also, quite importantly, includes adapting to different cultures, thereby avoiding an unnecessary process of semantic standardization.

3 Semantic Web Architecture for Language Technology

It is to be expected that semantic web methodologies (ontology-based metadata, web services) and standards (RDF, OWL) will be used in the specification of web-based, standardized language resources – data (corpora, lexicons, grammars) and tools – allowing for a distributed and widespread use of these resources in semantic web applications. Therefore, platforms will be needed for the discussion, implementation and dissemination of semantic web standards and protocols for the syntactic and semantic interoperability of language tools and resources across languages, cultures and applications.

This work should build on and reinforce previous and ongoing national, European and world-wide projects and initiatives in this area within language technology, e.g. ENABLER (European National Activities for Basic Language Resources), ICWLR (International Committee for Written Language Resources), IMDI (ISLE Metadata Initiative), INTERA (Integrated European Language Data Repository Area), MILE (Multilingual ISLE Lexical Entry), ISO/TC37/SC4, LT-World, OLAC (Open Language Archives Community), OLIF (Open Lexicon Interchange Format), while taking into account emerging (semantic) web standards as specified within W3C or industry, e.g. RDF/S, OWL, TopicMaps, Web Services Choreography Group, DAML-S, JXTA.

5 Conclusions

Effective acquisition, organization, processing, sharing, and use of the knowledge embedded in multimedia content as well as in information- and knowledge-based work processes plays a major role for competitiveness in the modern information society and for the emerging knowledge economy. However, this wealth of knowledge implicitly conveyed in the vast amount of available digital content is nowadays only accessible provided that considerable manual effort has been invested into its interpretation and semantic annotation, which is possible only for a small fraction of the available content. Therefore the major part of the implicit semantic knowledge is not taken into account by state-of-the-art information access technologies like search engines, which restrict their indexing activities to superficial levels, mostly the keyword level.

Multilinguality and multicultural expression are important aspects of human society. Texts and documents are - and will be - written in various native languages, but these documents are relevant even to non-native speakers. We could imagine bypassing the multilingual problem by focusing directly onto knowledge itself, rather than on language, but in fact, human knowledge is and will be encoded in language, and multilingual and multicultural aspects (culture as specific to countries, regions and nations, connected with language) will play an important role in establishing and maintaining such common understanding. The Semantic Web must represent and structure concepts in multilingual and multicultural ontologies, which can be obtained only by linking conceptual nodes with the various language specific lexical realizations.

Given these considerations, we are proposing a global research and development effort on establishing a distributed, standardized and semantically inter-

operable infrastructure of language resources and tools, which would enable a widespread integration of multilingual analysis tools into semantic web services and applications.

OntoGenie: Extracting Ontology Instances from WWW

Chintan Patel, Kaustubh Supekar, and Yugyung Lee

School of Computing and Engineering
University of Missouri-Kansas City
{copdk4, kss2r6, leeyu}@umkc.edu

Abstract. Web has become a tremendously huge information source on planet. However, the information is not machine perishable. Standardized Ontological representation of knowledge solves the problem as proposed by Semantic Web. One of the major challenges is to convert the information present on current Web into Ontologies for Semantic Web. We have developed a solution, OntoGenie, that parses the Web pages to create knowledge instances for a given Ontology using WordNet as a bridge, mapping between the Ontologies and the Web page terms. OntoGenie was tested over Ontologies available on the Semantic Web and some motivating results were obtained.

1 Introduction

Semantically enriched Web would allow leveraging intelligent applications such as semantic search, Semantic Web services and Semantic Grid. The knowledge in Semantic Web is encoded in webized way, as simple directed graphs [3]. Thus, Ontologies, representation of domain knowledge in Semantic Web, provide the explicit formalization and specification of the concepts and their corresponding relationships [2]. It should be noted that Ontologies have associated specific instances for the corresponding concepts. These instances contain the actual data that are being queried in knowledge based applications. Ontologies are largely developed manually by domain expert, filling in the instance data manually is an arduous task. It is infeasible to manually construct all instances corresponding to a concept defined in an Ontology.

In this paper, we focus on creating Ontology instances that can be automatically extracted from unstructured data on Web including plain text and HTML. Also, to accelerate the nurturing and growth of Semantic Web, there is a pressing need to develop tools that would provide smooth transition from current Web to Semantic Web. We have developed a tool, OntoGenie, that uses WordNet[1] to convert *unstructured data* from Web to *structured knowledge* for Semantic Web. The tool was developed as a part of ongoing BEE-SMART (A Natural Language Interface to Semantic Web) project at University of Missouri[2]. The architecture of the tool and the results obtained are discussed.

[1] http://www.cogsci.princeton.edu/ wn/
[2] http://sice527.ddns.umkc.edu/BeeSmart/

2 OntoGenie Functionality: What's your wish master?

The OntoGenie is a semi-automatic tool that takes as input domain ontologies and unstructured data from Web (plain text or HTML), and generates Ontology Instances (OI) for the given data. The tool uses the linguistic ontology, WordNet, as a bridge between domain ontologies and Web data.

Step 1: Map the concepts in a domain ontology into WordNet- Retrieve a concept C_d from domain ontology O_d and map it into a concept C_w in the WordNet ontology O_w. The mapping is performed by canonizing the English terms defining the Concepts (C_d and C_w). One important issue in this regard is that many terms in WordNet may map into a same concept from O_d. For example, the concept *University* in WordNet has more than one senses such as an 'educational institution' or a 'group of persons associated by some common tie'.

Step 2: Capture the terms occurring in Web pages- OntoGenie utilized the search service (Google Web service[3]) and the directory service (dmoz directory[4]) to retrieve Web pages for a particular domain. The web pages are parsed word by word, each word W_i is canonized and compared with the C_w present in the WordNet. Interestingly, we can visualize a connection among the Ontology Concepts (C_d), the WordNet Concepts(C_w) and the Web page terms (W_i). Consider the Ontology in Figure 1: the concept *Country* (C_d) in the domain ontology O_d is mapped to similar concept *Country* (C_w) in the WordNet ontology O_w in Step 1 and then during Step 2, the Web term *USA* (W_i) is mapped into a hyponym of *Country* C_w which has already being mapped to *Country* C_d.

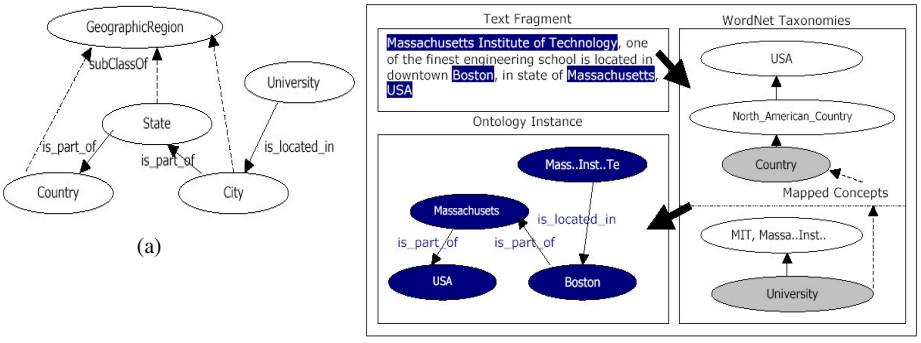


Fig. 1. (a) University Ontology Excerpt (b) Flow of OntoGenie Algorithm

Step 3: Discover relationships- Once the mappings are accomplished for a Web page, we discover the relationship that holds between the instances of the concepts extracted. Conventionally, the task of discovering relationships was

[3] http://www.google.com/apis/
[4] http://dmoz.org/

done via morphologically determining the verbs and the relationships to noun [1]. The approach works for simple *toy* cases, but fails practically in real world cases, dealing with large amount of ontological instances. We propose to use a simple approach using *principle of locality* (δ), the idea is to flexibly assume a set of concepts discovered in predetermined locus around the concepts to be related. To better understand the idea, consider the Ontology as a graph, with Concepts represented as nodes and the Relationships as links. The distance between Concepts in the set can be defined as number of links encountered traversing the links between the Concepts (we assume the shortest path).

Consider an example, as described in Figure 1b, an instance of University MIT and an instance of $Country$, we can assume a relationship to hold between them. It should be noted however that if the instances of intermediate nodes are unknown (e.g., *State* in this case), we still consider them as blank nodes. Such blank nodes can be filled on while scanning other Web pages for the given domain. The purpose of incorporating the principle of locality is to increase the recall by discovering largely disconnected knowledge instances and then linking them by information discovered from other pages.

3 OntoGenie Implementation and Results

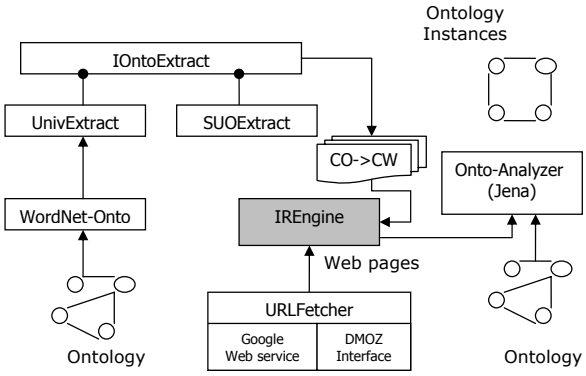


Fig. 2. OntoGenie Implementation Framework

The architecture of OntoGenie has been designed to exploit the functionality provided by the existing available tools. Figure 2 shows the implementation details for the OntoGenie framework. The OntoGenie implementation interacts with the Java WordNet and Jena APIs for Ontological and Web data parsing, computing locality-based distance between concepts, and creating Ontology instances. To disambiguate the Concept mappings to WordNet, as mentioned in Step 1, we have developed a graphical user interface for a domain expert to select the right sense for the automatically discovered mappings. We used KAON as our backend data store for crawled Ontologies. To interface Google Web service,

we used Java Web Services Developer Pack[5] (JWSDP). One of the noteworthy idea being incorporated is providing an abstract Interface *IOntoExtract*, wherein we can develop different plugins to test variety of mappings. For example, SUO[6] was mapped to WordNet within the OntoGenie framework. Similarly, with the component URLFetcher, one can add variety of interfaces to retrieve web pages (Google web services and DMOZ URL extractor were used in OntoGenie)

We tested the OntoGenie framework with University Ontology[7] and extracted the university related Web pages[8]. The OntoGenie has successfully discovered Knowledge Instances from the Web. Table 1 shows one of the RDF instance being discovered for the University Ontology. The excerpts says that **Librarian** is an instance of the concept *Person* and is the member of an *Organization* whose instance is **Library**.

<rdf:Description about= "http://tempuri.com/15univs.daml#Librarian">
<rdf:type resource= "UNIVURI#Person"/>
<km:member rdf:resource= "http://tempuri.com/km#Library"/>
</rdf:Description>
<rdf:Description about= "http://tempuri.com/km#Library">
<rdf:type resource= "UNIVURI#Organization"/>
</rdf:Description>
UNIVURI = http://www.cs.umd.edu/projects/plus/DAML/onts/cs1.0.daml

Table 1. Experimental Results

4 OntoGenie Conclusions: Getting back into Lamp!

We presented a simple, practical and implemented framework, OntoGenie that solves the highly critical and important problem of discovering Knowledge instances from Web. OntoGenie is based on creating mappings from Ontology to Web page terms using WordNet as a effective bridge. We showed implementation details and a glimpse of the results obtained.

References

1. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam and S. Slattery, Learning to Extract Symbolic Knowledge from the World Wide Web, Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98).
2. T.R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, Proceedings of International Workshop on Formal Ontology, Padova, Italy, 1993.
3. Tim Berners Lee, Semantic Web Roadmap, http://www.w3.org/DesignIssues/Semantic.htm

[5] http://java.sun.com/webservices/webservicespack.html
[6] http://suo.ieee.org/
[7] http://www.cs.umd.edu/projects/plus/DAML/onts/cs1.0.daml
[8] http://www.mit.edu:8001/people/cdemello/univ-full.html