# 3 Semantic Annotations and Retrieval: Manual, Semiautomatic, and Automatic Generation

*Kalina Bontcheva · Hamish Cunningham*
University of Sheffield, Sheffield, UK

John Domingue, Dieter Fensel & James A. Hendler (eds.), *Handbook of Semantic Web Technologies*,
DOI 10.1007/978-3-540-92913-0_3, © Springer-Verlag Berlin Heidelberg 2011

**Abstract:** The semantic annotation of textual Web content is key for the success of the Semantic Web. This entry reviews key approaches and state-of-the-art systems, as well as drawing conclusions on outstanding challenges and future work.

First, the problem of semantic annotation is defined and distinguished from other related research fields. Manual annotation tools are discussed next in the context of key requirements, such as support for diverse document formats, multiple ontologies, and collaborative, Web-based annotation.

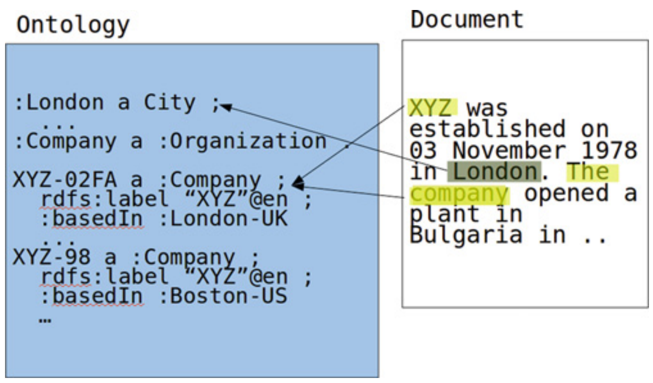
Next, the entry discusses ontology-oriented, semiautomatic, and automatic systems, which typically target ontologies as their output format, but do not use them as a knowledge resource during semantic analysis. Then a number of more advanced ontology-based semantic annotation approaches are presented and compared to one another. Particular emphasis is on scalability (i.e., the ability to process millions of documents) and customization (i.e., how easy it is to adapt these systems to new domains and/or ontologies).

The semantic retrieval of documents enables users to find all documents that mention one or more instances from the ontology and/or relations. The queries can also mix free-text keywords, not just the annotations. Here different types of retrieval tools are reviewed, some of which provide document browsing functionality as well as search refinement capabilities. The entry then provides in-depth examples of three semantic annotation applications: the GATE framework, News Collector, and large-scale patent processing. Future issues to be addressed are making use of linked data, dealing with large-scale, highly ambiguous ontologies, multilinguality, lexicalization of ontologies, and from an implementational perspective, semantic annotation as a service.

## 3.1 Scientific and Technical Overview

The Semantic Web is about adding a machine-tractable, repurposeable layer that complements the "traditional" Web of natural language hypertext. An important aspect of the World Wide Web is that it has been based largely on human-written materials, and in making the shift to the next-generation knowledge-based Web, human language will remain key. One particular example of the continuing importance of human language content on the Web comes from the success of Web 2.0 and social media. For instance, the growth in Twitter alone between 2008 and 2009 was over 1,000% and it is projected that by 2010, around 10% of all Internet users will be publishing content on Twitter. At the same time, there are over 70 million blogs and the average Facebook user has around 160 connections, many of whom are posting content in natural language on a daily basis. These users are also publishing other media online, such as photos and videos but these are outside the scope of this entry.

In the knowledge management context, Gartner reported (http://www3.gartner.com/DisplayDocument?id = 379859) that more than 95% of human-to-computer information input involves textual language and this trend will remain stable. They also report that by 2012, taxonomic and hierarchical knowledge mapping and indexing will be prevalent in



```
:London a City ;
  ...
:Company a :Organization .

XYZ-02FA a :Company ;
   rdfs:label "XYZ"@en ;
   :basedIn :London-UK
  ...
XYZ-98 a :Company ;
   rdfs:label "XYZ"@en ;
   :basedIn :Boston-US
   …
```

```
XYZ was
established on
03 November 1978
in London. The
company opened a
plant in
Bulgaria in ..
```

**Fig. 3.1**

**Semantic annotation example**

almost all information-rich applications. There is a tension here: between the increasingly rich semantic models in Semantic Web systems on the one hand and the continuing prevalence of human language materials on the other (see Ontologies and the Semantic Web, for an introduction to ontologies and the Semantic Web).

The process of tying semantic models and natural language together is referred to as *semantic annotation*. This process may be characterized as the dynamic creation of interrelationships between *ontologies* and unstructured and semi-structured documents in a bidirectional manner. From a technological perspective, semantic annotation is about annotating in texts all mentions of concepts from the ontology (i.e., classes, instances, properties, and relations), through metadata referring to their URIs in the ontology. Approaches that only enhance the ontology with new instances derived from the texts are typically referred to as *ontology population*.

A semantic annotation example is shown in *Fig. 3.1*, where the strings "XYZ" and "the company" are marked as referring to the instance with URI XYZ-02FA, which is of class company. From an implementational perspective, the semantic annotation task is often broken down into two main phases: ontology-based lookup and reference disambiguation (see *Fig. 3.2*). Ontology-based lookup is concerned with identifying all candidate mentions of concepts from the ontology. In this example, there are two candidates that match the string XYZ, based on their RDF labels: XYZ-02FA and XYZ-98. The reference disambiguation step then uses contextual information from the text as well as knowledge from the ontology to disambiguate the mentions to the correct ontology concept. In this example, the text mentions London and in the ontology, XYZ-02FA is the candidate company established in London.

Some semantic annotation systems also perform ontology population (see *Fig. 3.2*), that is, in addition to annotating the documents with respect to an ontology, they also enrich the ontology itself with new instances not already present in the ontology. For example, if a new British prime minister comes to power and a system is annotating news documents, then it can discover the new prime minister's name from the incoming

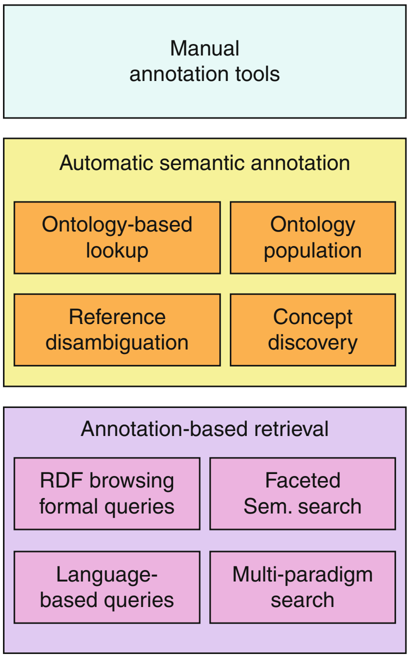


**Fig. 3.2**
**Semantic annotation and retrieval tasks and approaches**

articles. It must be noted that ontology population is a much harder task than ontology-based lookup and reference disambiguation, since it can introduce noisy, unreliable information in the ontology. An even more challenging problem is new concept discovery where a system can also learn new ontological classes and relations. The latter is typically carried out in a separate step, where domain experts can check the quality and validity of the newly discovered facts, prior to placing them in the ontology.

Semantic annotation can be performed manually, automatically, or semiautomatically, that is, first an automatic system creates some annotations and these are then post-edited and corrected by human annotators. Also, by definition all annotations are tied to one or more ontologies. Therefore, if an ontology changes or needs to be substituted by a different ontology, then all or some of the semantic annotation of the documents will need to be redone. Consequently, ontology evolution and the size of textual content on the Web make manual annotation infeasible in most cases, apart from very limited domains and applications. It is used primarily as means for checking the quality of the automatic methods, as well as for estimating the effort required for semiautomatic annotation.

*Information Extraction* (IE), a form of natural language analysis, is becoming a central technology for bridging the gap between unstructured text and formal knowledge expressed in ontologies. Ontology-Based IE (OBIE) is IE that is adapted specifically for the semantic annotation task. One of the important differences between traditional IE and OBIE is in the use of a formal ontology as one of the system's inputs and as the target output. Some researchers (e.g., [1]) call ontology-based any system that specifies its outputs with respect to an ontology, however, in general, if a system only has a mapping between the IE outputs and the ontology, this is not sufficient and therefore, such systems should be referred as *ontology-oriented*.

Another distinguishing characteristic of the ontology-based IE process is that it not only finds the (most specific) class of the extracted entity, but also identifies it by linking it to its semantic description in the instance base, typically via a URI. This allows entities to be traced across documents and their descriptions to be enriched during the IE process. In practical terms, this requires automatic recognition of named entities, terms, and relations and also coreference resolution both within and across documents. These more complex algorithms are typically preceded by some linguistic preprocessing (tokenization, Part-Of-Speech (POS) tagging, etc.).

Irrespective of the techniques used, the semantic annotation of textual content enables *semantic-based document retrieval* (see *Fig. 3.2*). This task is a modification of classical Information Retrieval (IR), but documents are retrieved on the basis of relevance to ontology concepts, as well as words. Nevertheless the basic assumption is quite similar – a document is characterized by the bag of tokens constituting its content, disregarding its structure. While the basic IR approach considers the word stems as tokens, there has been considerable effort for the last decade toward using word-senses or lexical concepts (see [2, 3]) for indexing and retrieval. The semantic annotations can be regarded as a special kind of token to be indexed and retrieved. With respect to techniques, work on semantic-based document retrieval is significantly less advanced than that on semantic annotation techniques, largely because the latter are enablers of the former. In addition, sufficiently scalable semantic repositories have only recently become available (for further details on semantic repositories see Storing the Semantic Web: Repositories).

Semantic annotation is relevant in many application contexts, for example, knowledge management (see Knowledge Management in Large Organizations), (eBusiness), and (eScience) (see the eponymous chapters in this handbook), and many large-scale implemented systems are already deployed and used on a daily basis.

## 3.1.1 Encoding Semantic Annotations

The first issue that needs to be addressed by any semantic annotation system is how to encode the annotations in documents. Some commonly used approaches are:

– As inline markup within the document's text content, with URIs pointing to the ontology (see *Fig. 3.13*)
– As RDF markup attached to the start/end of the document (see *Fig. 3.3*)
– As standoff RDF markup pointing to the document, but stored in a separate file and/or loaded within a semantic repository (see *Fig. 3.10*)

The trade-offs between these three approaches are several. First, representing each annotation inline, on each occurrence of the target instance/class has the advantage over

```
<!--ONTOMAT-ANNOTATION-BEGIN<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:iswc="http://annotation.semanticweb.org/2004/iswc#"
    ...
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <owl:Ontology rdf:about="http://www.dcs.shef.ac.uk/~kalina/index.html#">
    <owl:imports rdf:resource="http://annotation.semanticweb.org/ontologies/cream/ontomat#"/>
    <owl:imports rdf:resource="http://annotation.semanticweb.org/2004/iswc#"/>
  </owl:Ontology>
  <iswc:Organization rdf:about="http://www.dcs.shef.ac.uk/~kalina/index.html#University_of_Sheffield">
    <rdfs:label> University of Sheffield</rdfs:label>
    <iswc:has_affiliate>
      <iswc:Person rdf:about="http://www.dcs.shef.ac.uk/~kalina/index.html#Kalina_Bontcheva">
        <iswc:phone>(+44 - 114) 222 1930</iswc:phone>
        <iswc:fax>(+44 - 114) 222 1810</iswc:fax>
        <rdfs:label>Kalina Bontcheva</rdfs:label>
      </iswc:Person>
    </iswc:has_affiliate>
  </iswc:Organization>
...
</rdf:RDF>
-->
```

**Fig. 3.3**

**Example RDF markup attached to a semantically annotated document**

the other two representations when a system needs to retrieve the specific place(s) within documents where this class/instance is mentioned. The example shown in *Fig. 3.13* shows how all references to George Bush are annotated accordingly. In comparison, the other two representations only encode the fact that certain instances are mentioned in a particular document, but it is not possible to retrieve examples of where exactly these occur in the text. This approach makes the semantic annotation task considerably easier, since annotators only need to annotate only one of the mentions. Consequently, this also makes data storage and retrieval requirements smaller.

When comparing the second representation (RDF markup appended to the original document) to the third (RDF markup in a separate file), the choice depends on whether it is feasible to modify the document content itself. For an introduction to RDF, see Semantic Annotation and Retrieval: RDF.

## 3.1.2 Manual Semantic Annotation

Frameworks and user interface tools for manual semantic annotation need to address several challenges:

- First, as discussed above, they need to support references to ontology concepts via URIs.
- Second, given that manual annotation is time consuming, the tools should ideally be collaborative and also Web-based to enable distributed teams of annotators to share the work.

– Third, the tools need to go beyond annotation of classes and instances, and support also annotation of property and relation values.
– Fourth, the tools need to support annotation with respect to multiple ontologies and also scale well for large ontologies with many classes and relations. As discussed in [4], the annotation of relationships is significantly more time consuming for users and therefore a suitably supportive GUI is required.
– Last, but not least, manual annotation tools need to support multiple document formats, going beyond HTML toward PDF, XML, images (e.g., PNG, JPEG), and video.

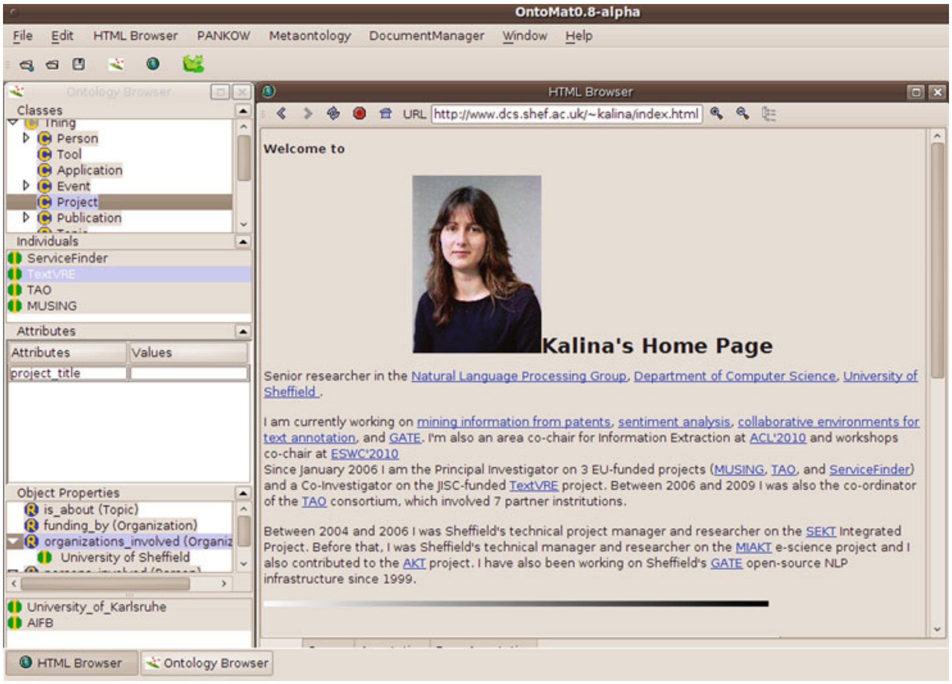
Next, several state-of-the-art manual annotation tools are discussed in the context of these challenges. For a description of some older systems please refer to [5].

A comprehensive semantic annotation framework is CREAM [4]. It not only addresses the requirements listed above, but it also provides a document editor that supports the creation of semantic annotations as an integral part of document authoring. Another distinguishing feature is the RDF crawler, which collects relevant entities from already published Semantic Web RDF data and makes these available to the human annotators, so they can reuse already existing instances, instead of creating new ones. CREAM also allows for the integration of automatic tools to bootstrap the manual annotation process (discussed in the following section).

CREAM's manual annotation editor is OntoMat Annotatizer (see *Fig. 3.4*) and it runs in a Web browser. There is an ontology guidance and fact browser, which allows users to expand the ontology with new data, for example, add a new instance. Document-based annotation is carried out by selecting parts of the text and then dropping them on the desired ontology class or, once a class has been chosen, in the property template for that class, in order to instantiate property values (e.g., a person's name or their date of birth). The example in *Fig. 3.4* shows a Web page annotated with people, projects, and organizations. However, although the new instances and annotations were created by first selecting them in the text, the mentions of these in the text itself are not highlighted, due to the fact that CREAM uses RDF triples, which are independent of the text itself (see *Fig. 3.3*).

Another manual semantic annotation editor for Web pages, similar to OntoMat, is SMORE (http://www.mindswap.org/2005/SMORE/) (see *Fig. 3.5*). It also integrates the SWOOP (http://www.mindswap.org/2004/SWOOP/) ontology editor. SMORE supports ontology navigation in order to select classes and properties and create triples to be added to the HTML pages. It also verifies the domain and range constraints on annotations to detect inconsistencies.

The W3C Annotea annotation framework and its extensions (e.g., [6]) support collaborative semantic annotation of documents accessible over the Internet, in multiple document formats, for example, HTML, PDF, images, and video. The framework uses RDF to model annotations as a set of statements. Annotations range from simple text comments, through hyperlinks, to controlled vocabulary statements (e.g., WordNet) and ontologies. As discussed in [6], annotations with respect to ontologies are modeled



Welcome to

Kalina's Home Page

Senior researcher in the Natural Language Processing Group, Department of Computer Science, University of Sheffield .

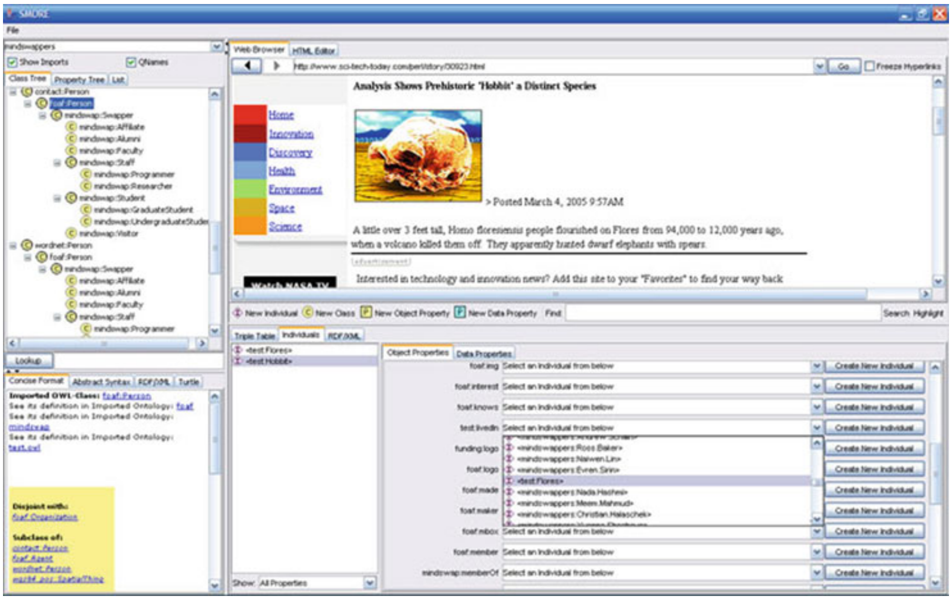
I am currently working on mining information from patents, sentiment analysis, collaborative environments for text annotation, and GATE. I'm also an area co-chair for Information Extraction at ACL'2010 and workshops co-chair at ESWC'2010

Since January 2006 I am the Principal Investigator on 3 EU-funded projects (MUSING, TAO, and ServiceFinder) and a Co-Investigator on the JISC-funded TextVRE project. Between 2006 and 2009 I was also the co-ordinator of the TAO consortium, which involved 7 partner institutions.

Between 2004 and 2006 I was Sheffield's technical project manager and researcher on the SEKT Integrated Project. Before that, I was Sheffield's technical manager and researcher on the MIAKT e-science project and I also contributed to the AKT project. I have also been working on Sheffield's GATE open-source NLP infrastructure since 1999.

**Fig. 3.4**

**OntoMat manual annotation tool**



Imported OWL-Class: foaf:Person
See its definition in Imported Ontology: foaf
See its definition in Imported Ontology: mindswap
See its definition in Imported Ontology: test.owl

Disjoint with:
foaf:Organization

Subclass of:
contact:Person
foaf:Agent
wordnet:Person
wgs84_pos:SpatialThing

Analysis Shows Prehistoric 'Hobbit' a Distinct Species

> Posted March 4, 2005 9:57AM

A little over 3 feet tall, Homo floresiensis people flourished on Flores from 94,000 to 12,000 years ago, when a volcano killed them off. They apparently hunted dwarf elephants with spears.

[advertisement]

Interested in technology and innovation news? Add this site to your "Favorites" to find your way back

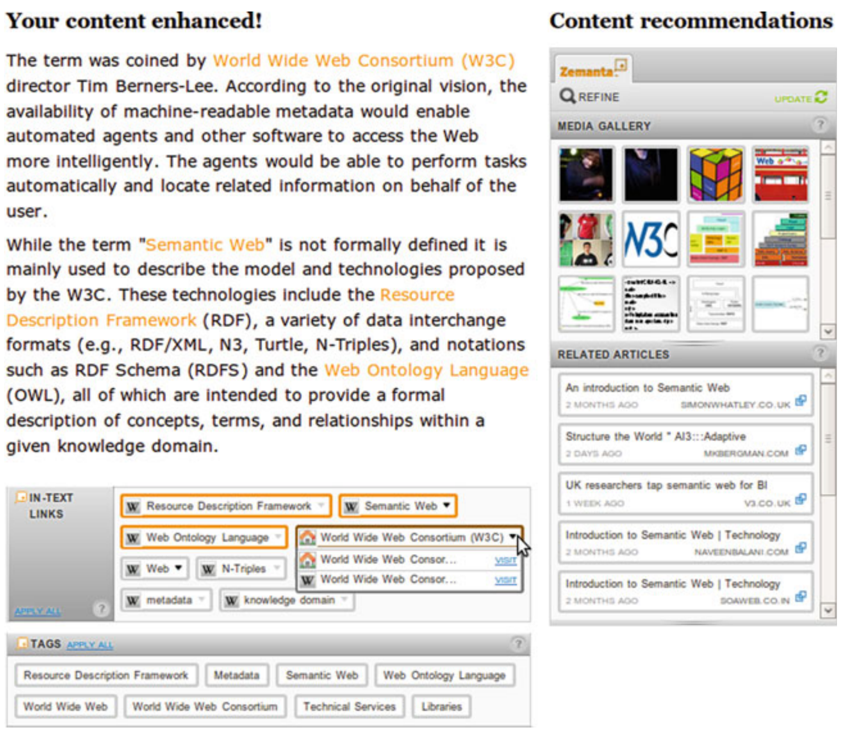
**Fig. 3.5**

**SMORE manual annotation UI**

through reification in order to support provenance, that is, information on who annotated what. The problem with reification, however, is that it is computationally expensive. The authors have therefore proposed to investigate named graphs in future work as a less expensive way to represent semantic annotations.

Zemanta (http://www.zemanta.com) is an online annotation tool for blog and e-mail content, which helps users insert tags and links through recommendations. ➲ *Figure 3.6* shows an example text and the recommended tags, potential in-text link targets (e.g., the W3C Wikipedia article and the W3C home page), and other relevant articles. It is then for the user to decide which of the tags should apply and which in-text link targets they wish to add. In this example, in-text links have been added for the terms highlighted in orange, all pointing to the Wikipedia articles on the respective topics.

There are also a number of multimedia semantic annotation tools, which are covered in more detail in ➲ Multimedia, Broadcasting, and eCulture. To take just one example, PhotoStuff [7] is an image annotation tool, which supports semantic annotation of images and regions of images with respect to OWL and RDFS ontologies. It defines an image region ontology (http://www.mindswap.org/2005/owl/digital-media), which has a set of useful concepts for image annotation. The semantic annotation interface is based


**Your content enhanced!**

The term was coined by World Wide Web Consortium (W3C) director Tim Berners-Lee. According to the original vision, the availability of machine-readable metadata would enable automated agents and other software to access the Web more intelligently. The agents would be able to perform tasks automatically and locate related information on behalf of the user.

While the term "Semantic Web" is not formally defined it is mainly used to describe the model and technologies proposed by the W3C. These technologies include the Resource Description Framework (RDF), a variety of data interchange formats (e.g., RDF/XML, N3, Turtle, N-Triples), and notations such as RDF Schema (RDFS) and the Web Ontology Language (OWL), all of which are intended to provide a formal description of concepts, terms, and relationships within a given knowledge domain.

IN-TEXT LINKS: Resource Description Framework; Semantic Web; Web Ontology Language; World Wide Web Consortium (W3C) — World Wide Web Consor... VISIT; World Wide Web Consor... VISIT; Web; N-Triples; metadata; knowledge domain

TAGS: Resource Description Framework; Metadata; Semantic Web; Web Ontology Language; World Wide Web; World Wide Web Consortium; Technical Services; Libraries

**Content recommendations**

RELATED ARTICLES

An introduction to Semantic Web — 2 MONTHS AGO — SIMONWHATLEY.CO.UK
Structure the World " AI3:::Adaptive — 2 DAYS AGO — MKBERGMAN.COM
UK researchers tap semantic web for BI — 1 WEEK AGO — V3.CO.UK
Introduction to Semantic Web | Technology — 2 MONTHS AGO — NAVEENBALANI.COM
Introduction to Semantic Web | Technology — 2 MONTHS AGO — SOAWEB.CO.IN

**Fig. 3.6**
**Zemanta's online tagging demo**

on forms, which provide slots for all properties of the chosen class and the user can then specify the values. For example, if the astronaut class is chosen for a given part of an image, then the form is populated with properties such as date of birth, education, and employer. The manual RDF annotations can then be uploaded into a semantic portal where they are published and made available for semantic searches (see Sect. 3.1.5). Provenance in this case is modeled at a file level, rather than annotation level, that is, each RDF file with semantic annotations is tagged with its creator name, description, and time stamp. On the one hand, this is a far more coarse-grained provenance model, but on the other, it is more computationally efficient.

In summary, while manual semantic annotation can be feasible in limited domains or through involving multiple annotators over the Web, it is in general considered too expensive to carry out without any automation. Consequently, the next section introduces semiautomatic and automatic approaches, many of which have been combined already with manual annotation.

## 3.1.3 Automatic and Semiautomatic Annotation

As discussed in the introduction, there are a number of ontology-oriented semantic annotation systems, which, unlike ontology-based ones, do not incorporate ontologies into the semantic analysis, but either use them as a bridge between the linguistic output and the final annotation (as with AeroDAML) or rely on the user to provide the relevant information through manual annotation (as with the Amilcare-based tools).

Information Extraction (IE) is one of the most commonly used techniques for (semi-) automatic semantic annotation. For example, when annotating information about companies, key information to be identified would be the company address, contact phone, fax numbers, and e-mail address, products and services, members of the board of directors, and so on. The field of information extraction has been driven by two major US international evaluation programs, from 1987 until 1997, the Message Understanding Conferences [8, 9] and since 2000, the Automatic Content Extraction Evaluation (ACE) [10].

The main tasks carried out during information extraction are:

- Named entity recognition, which consists of the identification and classification of different types of names in text
- Coreference resolution, which is the task of deciding if two linguistic expressions refer to the same entity in the discourse
- Relation extraction, which identifies relations between entities in text

Information extraction usually employs the following natural language processing components: Part-Of-Speech (POS) taggers, morphological analyzer, named entity recognizers, full (or shallow) parsing, and semantic interpretation. These linguistic processors are generally available (e.g., in language processing frameworks such as

GATE [11]), although some may require domain adaptation. For example, while a Parts-Of-Speech tagger can be reused mostly as is, a named entity recognizer would usually need adaptation to new application domains.

There are two main classes of approaches to information extraction:

1. Rule-based systems which are built by language engineers, who design lexicons and rules for extraction.
2. Machine-learning systems that are trained to perform one or more of the IE tasks. Learning systems are given either an annotated training corpus (i.e., supervised machine learning) or unannotated corpus together with a small number of seed examples (i.e., unsupervised or lightly supervised methods).

The advantages of rule-based approaches are that they do not require training data to create (although a small gold standard is needed for evaluation) and harness human intuition and domain knowledge. Depending on the lexical and syntactic regularity of the target domain, rule creation ranges from extremely fast (when few, clear patterns exist) to rather time consuming (if more ambiguities are present). Depending on the system design, some changes in requirements may be hard to accommodate. Since rule-based systems tend to require at least basic language processing skills, they are sometimes perceived as more expensive to create.

In comparison, machine-learning approaches typically require at least some human-annotated training data in order to reach good accuracy. While the cost per individual annotator is lower than the cost of language engineers, given the size of data needed, often more than one or two annotators are required. This raises the problem of inter-annotator agreement (or consistency), since the accuracy of the learnt models can be affected significantly by noisy, contradictory training data. However, getting training annotators to agree on their labels is again dependent on the complexity of the target annotations and could in itself be rather time consuming. Another potential problem could arise if the semantic annotation requirements change after the training data has been annotated, since this may require substantial re-annotation.

To summarize, both types of approaches have advantages and drawbacks and the choice of which one is more appropriate for a given application depends on the target domain, the complexity of the semantic annotations (including the size of the ontology), and the availability of trained human annotators and/or language engineers. Last but not least, there is no reason why one cannot have a hybrid approach, which uses both rules and machine learning.

Next, some representative semantic annotation systems are discussed with emphasis on their extraction components.

AeroDAML [12] is an annotation tool created by Lockheed Martin that applies IE techniques to automatically generate DAML annotations from Web pages. The aim is to provide naive users with a simple tool to create basic annotations without having to learn about ontologies, in order to reduce time and effort and to encourage people to semantically annotate their documents. AeroDAML links most proper nouns and common types of relations with classes and properties in a DAML ontology.

There are two versions of the tool: a Web-enabled version that uses a default generic ontology, and a client-server version that supports customized ontologies. In both cases, the user enters a URI (for the former) and a filename (for the latter) and the system returns the DAML annotation for the Web page or document. It provides a drag-and-drop tool to create static (manual) ontology mappings, and also includes some mappings to predefined ontologies.

*AeroDAML* consists of the AeroText IE system, together with components for DAML generation. A default ontology that directly correlates to the linguistic knowledge base used by the extraction process is used to translate the extraction results into a corresponding RDF model that uses the DAML + OIL syntax. This RDF model is then serialized to produce the final DAML annotation. The AeroDAML ontology is comprised of two layers: a base layer comprising the common knowledge base of AeroText, and an upper layer based on WordNet [13]. AeroDAML can generate annotations consisting of instances of classes such as common nouns and proper nouns, and properties, of types such as coreference, Organization to Location, Person to Organization.

*Amilcare* [14] is an adaptive IE system that has been integrated in several different annotation tools for the Semantic Web. It uses machine learning (ML) to learn to adapt to new domains and applications using only a set of annotated texts (training data). It has been adapted for use in the Semantic Web by simply monitoring the kinds of annotations produced by the user in training, and learning how to reproduce them. The traditional version of Amilcare adds XML annotations to documents (inline markup); the Semantic Web version (used by Melita – see below) leaves the original text unchanged and produces the extracted information as triples of the form < annotation, startPosition, endPosition > (standoff markup – see ❯ Sect. 3.1.1). This means that it is left to the annotation tool and not the IE system to decide on the format of the ultimate annotations produced.

In the Semantic Web version, no knowledge of IE is necessary; the user must simply define a set of annotations, which may be organized as an ontology where annotations are associated with concepts and relations. The user then manually annotates the text using some interface connected to Amilcare, as described in the following systems. Amilcare works by preprocessing the texts using GATE's IE system ANNIE [15], and then uses a supervised machine learning algorithm [16] to induce rules from the training data.

*Melita* [17] is an ontology-based tool for semantic annotation, which provides a mechanism for a user to interact with an IE system (Amilcare). It consists of two main parts: an ontology viewer and a document editor. The two most interesting features of Melita are that it enables the user to tune the IE system to provide different levels of proactivity, and to schedule texts to provide timeliness (i.e., learning with minimum delay). The annotation cycle follows two phases: manual annotation (training of the system) and active annotation (where the system takes over the annotation automatically). At some point, the system will start suggesting annotations to the user (active annotation) and the user can correct these as necessary. The system can suggest annotations as either reliable or unreliable, depending on its confidence level about that annotation. Reliable annotations need to be explicitly removed by the user, while unreliable annotations need to be explicitly added.

*MnM* [18] is a semantic annotation tool that provides support for annotating Web pages with semantic metadata. This support is semiautomatic, in that the user must provide some initial training information by manually annotating documents before the IE system (Amilcare) can take over. It integrates a Web browser, an ontology editor, and tools for IE, and has been described as “an early example of next-generation ontology editors” [18], because it is Web-based and provides facilities for large-scale semantic annotation of Web pages.

The philosophy behind MnM is that the semantic annotation of Web pages can, and should, be carried out by users without specialist skills in either language technology or knowledge engineering. It therefore aims to provide a simple system to perform knowledge extraction tasks at a semiautomatic level.

The five main steps to the underlying procedure are:

– The user browses the Web.
– The user manually annotates his chosen Web pages.
– The system learns annotation rules.
– The system tests the rules learnt.
– The system takes over automatic annotation, and populates ontologies with the instances found.

The ontology population process is semiautomatic and may require intervention from the user. First, it only deals with a predefined set of concepts in the ontology. Second, the system is not perfect and may miss instances in the text, or allocate them wrongly. Retraining can be carried out at any stage, however.

*S-CREAM* (Semiautomatic CREAtion of Metadata) [19] is a tool that provides a mechanism for automatically annotating texts, given a set of training data, which must be manually created by the user. It uses a combination of two tools: Onto-O-Mat, a manual annotation tool that implements the CREAM framework for creating relational metadata [20], and Amilcare.

As with the other Amilcare-based tools, S-CREAM is trainable for different domains, provided that the user creates the necessary training data. It essentially works by aligning conceptual markup (which defines relational metadata) provided by OntoMat with semantic markup provided by Amilcare. This problem is not trivial because the two representations may be very different. Relational metadata may provide information about relationships between instances of classes, for example that a certain hotel is located in a certain city. S-CREAM thus supports metadata creation with the help of a traditional IE system, and also provides other functionalities such as a Web crawler, a document management system, and a meta-ontology.

*Wrapper-based Data Extraction for Ontology Population*: Lixto [21] is a set of tools for writing wrappers that scrape Web pages and perform data extraction. As part of the REWERSE project these were used to build essentially an ontology population tool, which scrapes and syndicates information from publication pages. The output is an ontology of researchers and publications that is populated automatically from the Web pages. However, unlike all previous systems, the goal here is only ontology population, that is, the original Web content is not annotated semantically.

*Drawbacks of the Ontology-Oriented Approaches*: One of the problems with ontology-oriented annotation tools, such as those reviewed here, is that they do not provide the user with a way to customize the integrated language processing directly. While many users would not need or want such customization facilities, users who already have ontologies with rich instance data will benefit if they can make these data available to the IE components. However, this is not possible when traditional IE methods such as Amilcare are used, because they are not aware of the existence of the user ontology.

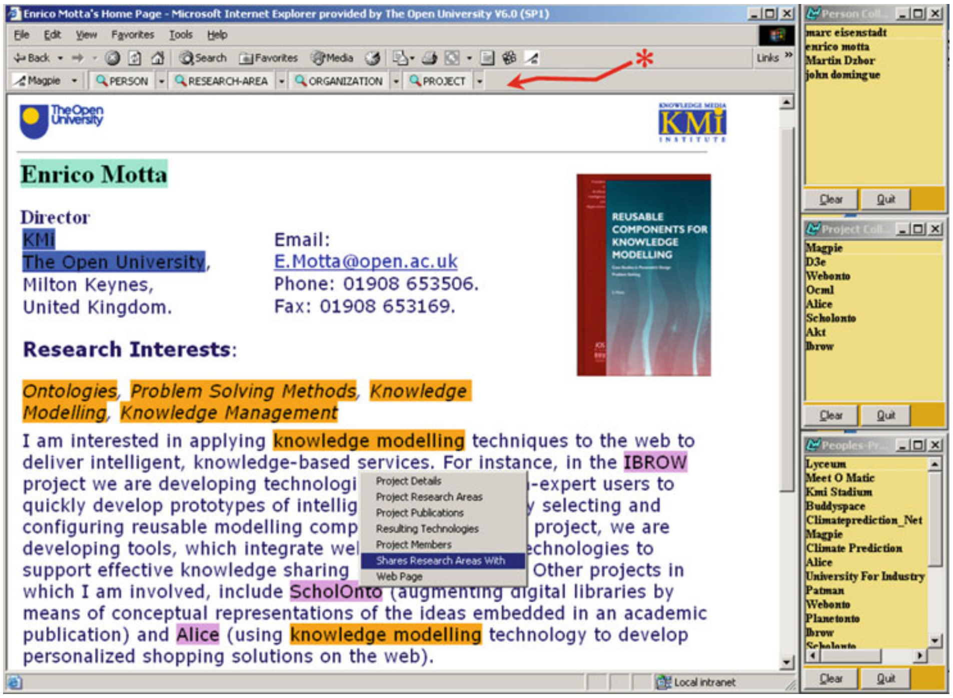
The more serious problem however, as discussed in the S-CREAM system [19], is that there is often a gap between the IE output annotations and the classes and properties in the user's ontology. The solution proposed by the developers was to write logical rules to resolve this. For example, an IE system would typically annotate London and UK as locations, but extra rules are needed to specify that there is a containment relationship between the two. However, rule writing of this kind is too difficult for most users and therefore ontology-based semantic annotation algorithms were developed, as they annotate directly with the classes and instances from the user's ontology.

*Magpie* [22] is a suite of tools that supports the interpretation of Web pages and "collaborative sense-making." It annotates Web pages with metadata in a fully automatic fashion and needs no manual intervention by matching the text against instances in the ontology (see *Fig. 3.7*). It automatically populates an ontology from relevant Web sources, and can be used with different ontologies. The principle behind it is that it uses an ontology to provide a very specific and personalized viewpoint of the Web pages the user wishes to browse. This is important because different users often have different degrees of knowledge and/or familiarity with the information presented, and have different browsing needs and objectives.

Another interesting aspect of Magpie is that it maintains a kind of "browsing history" in windows called *collectors*. Each collector shows the instances of a given concept that have been mentioned on the page or a list of related instances (e.g., people working on a given project, which were not mentioned in this page). The user can then click on these instances and browse their semantic data or create *semantic bookmarks* to retrieve this information later through semantic queries.

However, Magpie relies on a prespecified ontology, which makes the system domain-dependent. PowerMagpie [23] is an extension of the approach, so that it identifies automatically, at runtime, the most appropriate ontology to be used for annotation. PowerMagpie displays two panels. The first one is called "Entities" and lists key terms from the Web page, as well as the ontological entities they refer to and navigation to the places where they are mentioned in the text. The second panel is called "Ontologies" and displays the automatically found ontologies, which were deemed relevant to the given page.

In PowerMagpie, semantic annotation is performed first by identifying statistically the domain terms and then matching them up against candidate ontologies. The system first


Enrico Motta's Home Page - Microsoft Internet Explorer provided by The Open University V6.0 (SP1)
File Edit View Favorites Tools Help
Back Search Favorites Media Links
Magpie PERSON RESEARCH-AREA ORGANIZATION PROJECT
The Open University
KMi
REUSABLE COMPONENTS FOR KNOWLEDGE MODELLING
Project Details
Project Research Areas
Project Publications
Resulting Technologies
Project Members
Shares Research Areas With
Web Page
Local intranet
Person Coll...
Project Coll...
Peoples-Pr...
Clear
Quit

Enrico Motta

Director
KMi
The Open University,
Milton Keynes,
United Kingdom.

Email:
E.Motta@open.ac.uk
Phone: 01908 653506.
Fax: 01908 653169.

Research Interests:

Ontologies, Problem Solving Methods, Knowledge Modelling, Knowledge Management

I am interested in applying knowledge modelling techniques to the web to deliver intelligent, knowledge-based services. For instance, in the IBROW project we are developing technologi -expert users to quickly develop prototypes of intellig y selecting and configuring reusable modelling comp project, we are developing tools, which integrate wel echnologies to support effective knowledge sharing Other projects in which I am involved, include ScholOnto (augmenting digital libraries by means of conceptual representations of the ideas embedded in an academic publication) and Alice (using knowledge modelling technology to develop personalized shopping solutions on the web).

marc eisenstadt
enrico motta
Martin Dzbor
john domingue

Magpie
D3e
Webonto
Ocml
Alice
Scholonto
Akt
Ibrow

Lyceum
Meet O Matic
Kmi Stadium
Buddyspace
Climateprediction_Net
Magpie
Climate Prediction
Alice
University For Industry
Patman
Webonto
Planetonto
Ibrow
Scholonto

**Fig. 3.7**
**Magpie example**

used a TF*IDF term recognizer, but due to over-generation it was replaced by a call to the Yahoo! Term extraction service (http://developer.yahoo.com/search/content/V1/termExtraction.html). The key terms are then used to select dynamically one or more relevant ontologies. The matching is done on the basis of string similarity between the key terms and the labels of classes, instances, and properties in the ontology. Complex terms, for example, University of Sheffield, are matched as a whole, rather than matching university and Sheffield separately. However, it remains unclear how PowerMagpie would deal with named entity recognition and also with relation annotation.

PANKOW and OntoSyphon: The PANKOW system (Pattern-based Annotation through Knowledge on the Web) [24] exploits surface patterns and the redundancy on the Web to categorize automatically instances from text with respect to a given ontology. The patterns are phrases like: the < INSTANCE > <CONCEPT > (e.g., the Ritz hotel) and < INSTANCE > is a < CONCEPT > (e.g., Novotel is a hotel). The system constructs patterns by identifying all proper names in the text (using a Part-of-Speech tagger) and combining each one of them with each of the 58 concepts from their tourism ontology into a hypothesis. Each hypothesis is then checked against the Web via Google queries and the number of hits is used as a measure of the likelihood of this pattern being correct.

The system's best performance on this task in fully automatic mode is 24.9%, while the human performance is 62.09%. However, when the system is used in semiautomatic mode, that is, it suggests the top five most likely concepts and the user chooses among them, then the performance goes up to 49.56%.

The advantages of this approach are that it does not require any text processing (apart from POS tagging) or any training data. All the information comes from the Web. However, this is also a major disadvantage because the method does not compare the context in which the proper name occurs in the document to the contexts in which it occurs on the Web, thus making it hard to classify instances with the same name that belong to different classes in different contexts (e.g., Niger can be a river, state, country, etc.). On the other hand, while IE systems are more costly to set up, they can take context into account when classifying proper names.

Another system similar to PANKOW is OntoSyphon [1], which uses the ontology as the starting point in order to carry out Web mining to populate the ontology with instances. It uses the ontology structure to determine the relevance of the candidate instances. However, it does not carry out semantic annotation of documents as such.

There has also been work on populating ontologies specifically from tabular data from the Web, for example, the AllRight system [25]. The approach is based on clustering, table identification, and conflict detection.

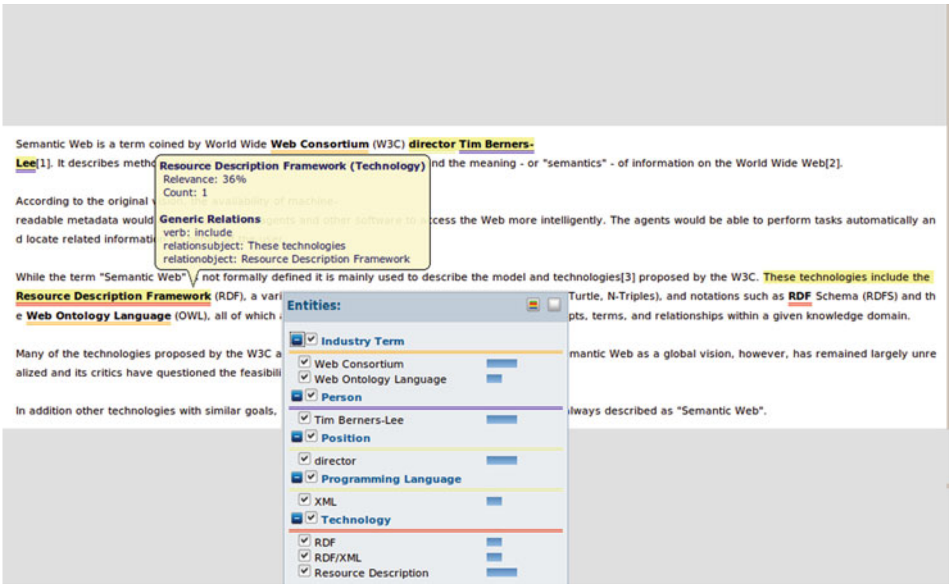
*Open Calais* is a commercial Web service provided by Thomson Reuters that carries out semantic annotation. At the time of writing, the target entities are mostly locations, companies, people, addresses, contact numbers, products, movies, etc. The events and facts extracted are those involving the above entities, for example, acquisition, alliance,and company competitor. The Calais OWL ontology is available at: http://www.opencalais.com/documentation/opencalais-web-service-api/calais-ontology-owl. ❯ *Figure 3.8* shows an example text annotated with some entities.

The Calais service also carries out limited entity disambiguation for companies (with respect to a proprietary database of public companies); locations (using Freebase); and electronics (using Shopping.com).

The entity annotations include URIs, which allow access via HTTP to obtain further information on that entity via linked data. Currently OpenCalais links to eight linked datasets, including DBPedia, Wikipedia, IMDB, and Shopping.com. These broadly correspond to the entity types covered by the ontology.

The main limitation of Calais comes from its proprietary nature, that is, users send documents to be annotated by the Web service and receive results back, but they do not have the means to give Calais a different ontology to annotate with or to customize the way in which entity extraction works.

*SemTag* [26] performs large-scale semantic annotation with respect to the TAP ontology (http://tap.stanford.edu/data/). It first performs a lookup phase annotating all possible mentions of instances from the TAP ontology. In the second, disambiguation phase, SemTag uses a vector-space model to assign the correct ontological class or to determine that this mention does not correspond to a class in TAP. The disambiguation is carried out by comparing the context of the current mention against the contexts of instances in TAP with compatible aliases, using a window of ten words either side of the mention.



Semantic Web is a term coined by World Wide **Web Consortium** (W3C) **director Tim Berners-Lee**[1]. It describes metho nd the meaning - or "semantics" - of information on the World Wide Web[2].

According to the original

readable metadata would cess the Web more intelligently. The agents would be able to perform tasks automatically an

d locate related informati

While the term "Semantic Web" not formally defined it is mainly used to describe the model and technologies[3] proposed by the W3C. These technologies include the **Resource Description Framework** (RDF), a vari Turtle, N-Triples), and notations such as **RDF** Schema (RDFS) and th

e **Web Ontology Language** (OWL), all of which pts, terms, and relationships within a given knowledge domain.

Many of the technologies proposed by the W3C a mantic Web as a global vision, however, has remained largely unre

alized and its critics have questioned the feasibili

In addition other technologies with similar goals, lways described as "Semantic Web".

**Resource Description Framework (Technology)**
Relevance: 36%
Count: 1

**Generic Relations**
verb: include
relationsubject: These technologies
relationobject: Resource Description Framework

**Entities:**

- ✓ **Industry Term**
  - ✓ Web Consortium
  - ✓ Web Ontology Language
- ✓ **Person**
  - ✓ Tim Berners-Lee
- ✓ **Position**
  - ✓ director
- ✓ **Programming Language**
  - ✓ XML
- ✓ **Technology**
  - ✓ RDF
  - ✓ RDF/XML
  - ✓ Resource Description

**Fig. 3.8**
**Calais results on part of the Semantic Web Wikipedia entry**

The TAP ontology, which contains about 65,000 instances, is very similar in size and structure to the KIM Ontology and KB (e.g., each instance has a number of lexical aliases). One important characteristic of both ontologies is that they are lightweight and encode only essential properties of concepts and instances. In other words, the goal is to cover frequent, commonly known and searched for instances (e.g., capital cities, names of presidents), rather than to encode an extensive set of axioms enabling deep, Cyc-style reasoning. As reported in [27], the heavyweight logical approach undertaken in Cyc is not appropriate for many NLP tasks.

The SemTag system is based on a high-performance parallel architecture – Seeker, where each node annotates about 200 documents per second. The demand for such parallelism comes from the big volumes of data that need to be dealt with in many applications and make automatic semantic annotation, the only feasible option. A parallel architecture of a similar kind is currently under development for KIM and, in general, it is an important ingredient of large-scale automatic annotation approaches.

*The KIM Semantic Annotation Platform* [28, 29] is an extendable platform for knowledge management that offers facilities for metadata creation, storage, and semantic-based search (see *Fig. 3.9* for details). It also includes a set of front ends for online use that offer multi-paradigm search and semantically enhanced browsing (see Sect. 3.1.5).

KIM uses the PROTON ontology (http://proton.semanticweb.org/) that contains around 250 classes and 100 properties. The classes cover entities (such as people, organizations, locations, products) and events. The core entities addressed are roughly equivalent to those covered by OpenCalais.

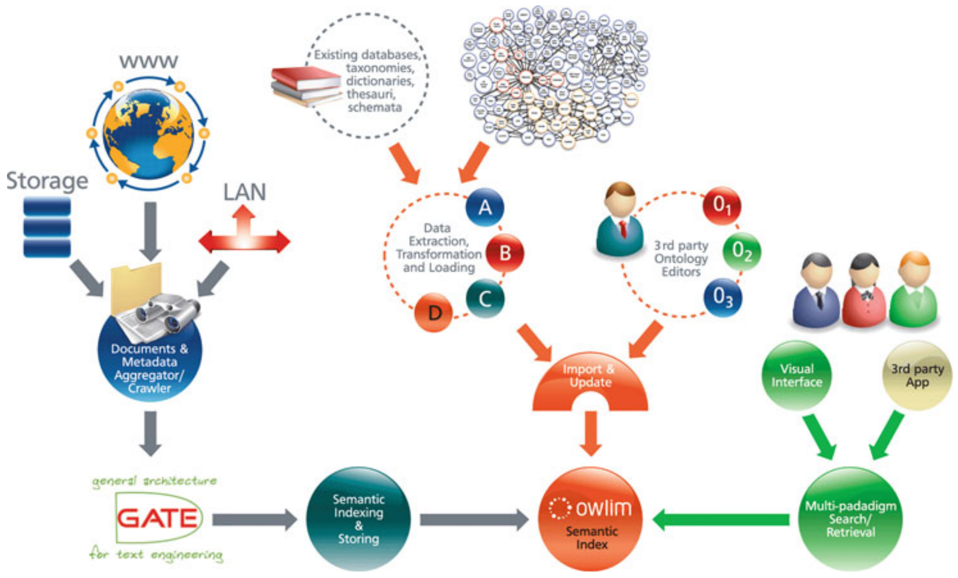

WWW
Storage
LAN
Existing databases, taxonomies, dictionaries, thesauri, schemata
Data Extraction, Transformation and Loading
A
B
C
D
3rd party Ontology Editors
O1
O2
O3
Documents & Metadata Aggregator/ Crawler
Import & Update
Visual Interface
3rd party App
general architecture
GATE
for text engineering
Semantic Indexing & Storing
owlim Semantic Index
Multi-padadigm Search/ Retrieval

**Fig. 3.9**
**KIM architecture**

The information extraction in KIM is based on the GATE framework [11]. The essence of KIM's semantic annotation is the recognition of named entities with respect to the KIM ontology. The entity instances all bear unique identifiers that allow annotations to be linked both to the entity type and to the exact individual in the instance base. For new (previously unknown) entities, new identifiers are allocated and assigned; then minimal descriptions are added to the semantic repository. The annotations are kept separately from the content, and an API for their management is provided.

The instance base of KIM is pre-populated with 200,000 entities of general importance that occur frequently in documents. The majority are different kinds of locations: continents, countries, cities, etc. Each location has geographic coordinates and several aliases (usually including English, French, Spanish, and sometimes the local transcription of the location name) as well as co-positioning relations (e.g., subRegionOf.) As previously shown by [30], IE systems need such data, because locations are difficult to recognize otherwise.

The difference between SemTag's TAP ontology and the KIM instance base is in the level of ambiguity. TAP has few entities sharing the same alias, while KIM has a lot more, due to its richer collection of locations.

At a conceptual level, KIM and SemTag differ significantly in their goal. Namely, SemTag aims only at accurate classification of the mentions that were found by matching the lexicalizations in the ontology. KIM, on the other hand, also aims to find *all mentions*, that is, coverage, as well as accuracy. The latter is a harder task because there tends to be a trade-off between accuracy and coverage. In addition, SemTag does not attempt to discover and classify new instances, which are not already in the TAP ontology. In other words, KIM performs two tasks, ontology population with new instances and semantic annotation, while SemTag performs only semantic annotation.

KIM can also use linked data ontologies for semantic annotation. At present it has been tested with DBPedia, Geonames, WordNet, Musicbrainz, Freebase, UMBEL, Lingvoj, and the CIA World Factbook (for further details on linked data see Semantic Annotation and Retrieval: Web of Data). Those datasets are preprocessed and loaded to form an integrated dataset of about 1.2 billion explicit statements. Forward-chaining is performed to materialize another 0.8 billion implicit statements, in accordance with the semantics of the ontologies used in the datasets.

Another important issue is extensibility, where not only the ontology can be replaced or extended, but also there is the option to customize or replace the semantic annotation application used within KIM. This can be any GATE-based semantic application, including machine learning, rule-based, or any other text-processing component integrated in GATE. The semantic annotator could also be provided by any other system, provided that it is wrapped and plugged into KIM via its API. In fact, KIM's extensibility with respect to new ontologies and semantic annotators is what makes it more powerful than ready-made services, such as OpenCalais.

*Ontology-Based Semantic Annotation via Hierarchical Learning*: The Hieron system [31] implements a hierarchical learning approach for semantic annotation, which uses the target ontology as an essential part of the annotation process, by taking into account the relations between concepts and instances in the ontology.

As discussed above, conventional IE uses labels that have no specific relation among each other, that is, they are treated as independent by the learning algorithms (e.g., Person, Location). However, concepts in an ontology are related to each other (at the very least through the subsumption hierarchy) and therefore it is beneficial to feed this knowledge into the OBIE algorithms. The Hieron system has explored two aspects of using the ontology structure for semantic annotation. First, it derives ontology-induced measures, which are then used by the learning algorithm to evaluate how well it is learning to annotate the target concepts. Second, the authors introduce the Perceptron-based learning algorithm Hieron, which has a mechanism to handle effectively hierarchical classification, as is required for semantic annotation.

The approach was evaluated on a corpus of 290 news articles annotated manually with respect to an ontology of 146 classes. The results demonstrate clearly the benefits of using knowledge from the ontology as input to the information extraction process.

## 3.1.4 Entity Disambiguation

Gruhl et al. [32] focus in particular on the disambiguation element of semantic annotation and examine the problem of dealing with highly ambiguous cases. Their approach first restricts the part of the ontology used for producing the candidates, in this case by filtering out all information about music artists not mentioned in the given text. Second, they apply lightweight language processing, such as POS tagging, and then use this information as input to a support vector machine classifier, which disambiguates on the basis of this information. The approach has been tested with the MusicBrainz ontology and a corpus of MySpace posts for three artists. While the ontology is very large (thus generating a lot of ambiguity), the texts are quite focused, which allows the system to achieve good performance. As discussed by the authors themselves, the processing of less focused texts, for example, Twitter messages or news articles, is likely to prove far more challenging.

The problem of person name disambiguation is a specific case of entity disambiguation, which has received attention in earlier work. For example, Aswani et al. [33] experimented with disambiguating author names in citations by using both contextual information (e.g., coauthor names) and additional evidence gathered from the Web, in combination with a similarity measure. Similar to Gruhl et al., the reported accuracy was very good, but again, the question remains open as to how well the approach will deal with other domains or text types.

Another approach to named entity disambiguation, called IdentityRank [34], was proposed in the context of the NEWS project. It tackles the problem of disambiguating named entities occurring in newspaper articles with respect to a news domain ontology. The algorithm exploits the metadata provided by news agencies, automatically detected named entities, and NewsCodes subject categories. It also takes into account the frequency of occurrence of entities in the last few days, as well as the frequency of occurrence in news with a given category. However, similar to the previously discussed approaches, IdentityRank has not been tested on other domains and applications and, thus, the general applicability of the approach outside the news domain remains unproven.

The IdRF framework for identity resolution [35] differs from the above work, since it exploits instead knowledge from the ontology, in order to determine whether a candidate mention from the text refers to a known instance in the ontology or a new one needs to be created. For efficiency reasons, the resolution process is divided into several steps. First is pre-filtering, which filters out the irrelevant parts of the ontology and forms a set of candidate entities. Next is the similarity measure stage, during which context from the text is compared against knowledge in the ontology to help with disambiguation. The last stage is called data integration and it determines whether to assert a new instance in the ontology or match the mention to an existing one, based on the similarity measures from the previous step. The potential drawback of this approach is that it requires the manual definition of the similarity metrics and pre-filtering criteria, which might prove complex as the size of the ontology grows.

### 3.1.5 Annotation Retrieval

Semantic annotations in documents enable users to find all documents that mention one or more instances from the ontology and/or relations. The queries can also mix free-text keywords, not just the annotations. Most retrieval tools provide also document browsing functionality as well as search refinement capabilities. Due to the fact that documents can have hundreds of annotations (especially if every concept mention in the document is annotated), annotation retrieval on a large document collection is a very challenging task.

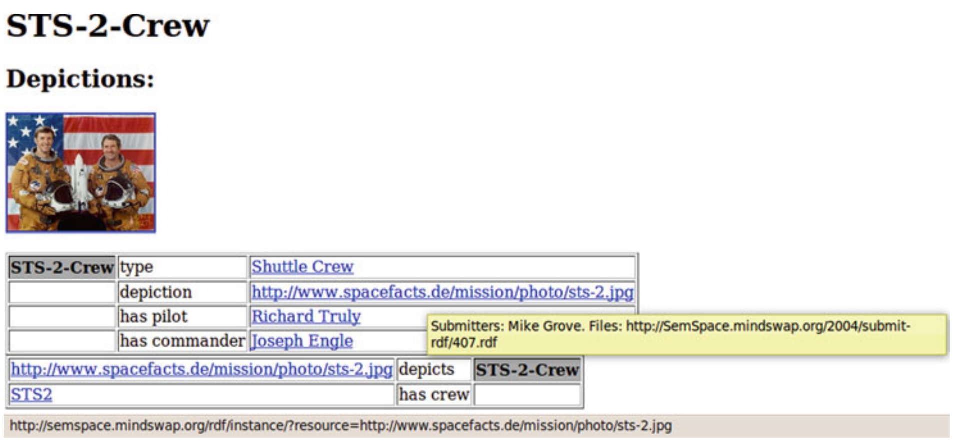
Annotation-based search and retrieval is different from traditional information retrieval, because of the underlying graph representation of annotations, which encode structured information about text ranges within the document. The encoded information is different from the words and inter-document link models used by Google and other search engines. In the case of semantic annotations, the case becomes even more complex, since they also refer to ontologies via URIs. While augmented full-text indexes can help with efficient access, the data storage requirements can grow exponentially with the cardinality of the annotation sets. Therefore different, more optimized solutions have been investigated.

The main difference from Semantic Web search engines, such as Swoogle [36], is the focus on annotations and using those to find documents, rather than forming queries against ontologies or navigating ontological structures. Similarly, semantic-based facet search and browse interfaces, such as /facet [37], tend to be ontology-based, whereas annotation-based facet interfaces (see KIM below) tend to hide the ontology and instead resemble more closely "traditional" string-based faceted search.

Next, several representative approaches are discussed.

*Browsing RDF Annotations*: The MINDSWAP SemPortal (http://www.mindswap.org/) publishes RDF annotations on the Web, for example, those created by PhotoStuff [7]. As can be seen in *Fig. 3.10*, the user can then browse these RDF instances via the portal and see any associated images and other documents. The provenance of the information is shown as a tooltip as shown in the example.

*Natural Language Interfaces*: These allow users to perform retrieval tasks using written or spoken language (e.g., English). The majority of work on natural language interfaces for



**STS-2-Crew**

**Depictions:**

| | | |
|---|---|---|
| **STS-2-Crew** | type | Shuttle Crew |
| | depiction | http://www.spacefacts.de/mission/photo/sts-2.jpg |
| | has pilot | Richard Truly |
| | has commander | Joseph Engle |

Submitters: Mike Grove. Files: http://SemSpace.mindswap.org/2004/submit-rdf/407.rdf

| | | |
|---|---|---|
| http://www.spacefacts.de/mission/photo/sts-2.jpg | depicts | **STS-2-Crew** |
| STS2 | has crew | |

http://semspace.mindswap.org/rdf/instance/?resource=http://www.spacefacts.de/mission/photo/sts-2.jpg

**Fig. 3.10**
**SemPortal: Instance browsing example**

the Semantic Web has focused on the problem of querying ontologies (e.g., [38–40]) or ontology authoring (e.g., [41, 42]). Using language-based queries for retrieving semantic annotations and associated documents is a somewhat different task, since the queries need to go beyond the ontology and into documents as well.

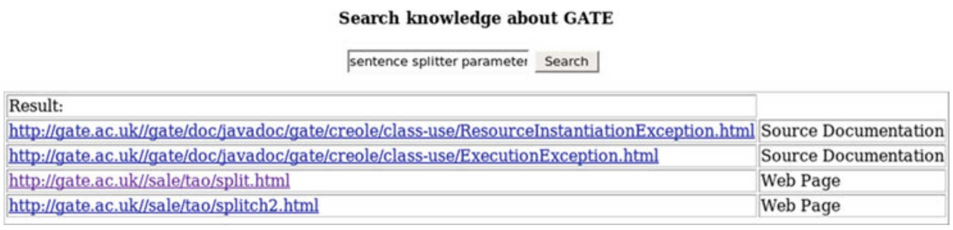
The QuestIO system [43], for example, has an ontology modeling document and the semantic annotations in them and uses it to help naive users to search through RDF annotations and get a list of matching documents back. The example domain is software engineering where over 10,000 different artifacts (software code, documentation, user manuals, papers, etc.) were annotated semantically with respect to a domain ontology. *Figure 3.11* shows a query where the user needs more information about the parameters of a particular component, called Sentence Splitter. The results are a list of document URLs that mention the query concepts. QuestIO implicitly interprets the query as a search for all documents discussing Sentence Splitter parameters.

QuestIO interprets the queries as follows. First it tries to match some or all of the contained words to ontology concepts. Then any remaining textual segments are used to predict property names and act as context for disambiguation. The sequence of concepts and property names can then be converted into a formal query that is executed against the semantically annotated documents. Throughout the process, metrics are used to score the possible query interpretations, allowing the filtering of low scoring options, thus reducing ambiguity and limiting the search space.

Another similar system is SemSearch [44], which is based on Sesame for indexing the semantics and Lucene for indexing the texts. Queries can be a combination of keywords (e.g., news) and connectors such as "and" and "or" (see *Fig. 3.12* for an example). The system performs semantic matching between words in complex queries and semantic entities by exploring different plausible combinations between the keywords. For longer queries this could compromise the performance of the search engine and more efficient strategies are needed.

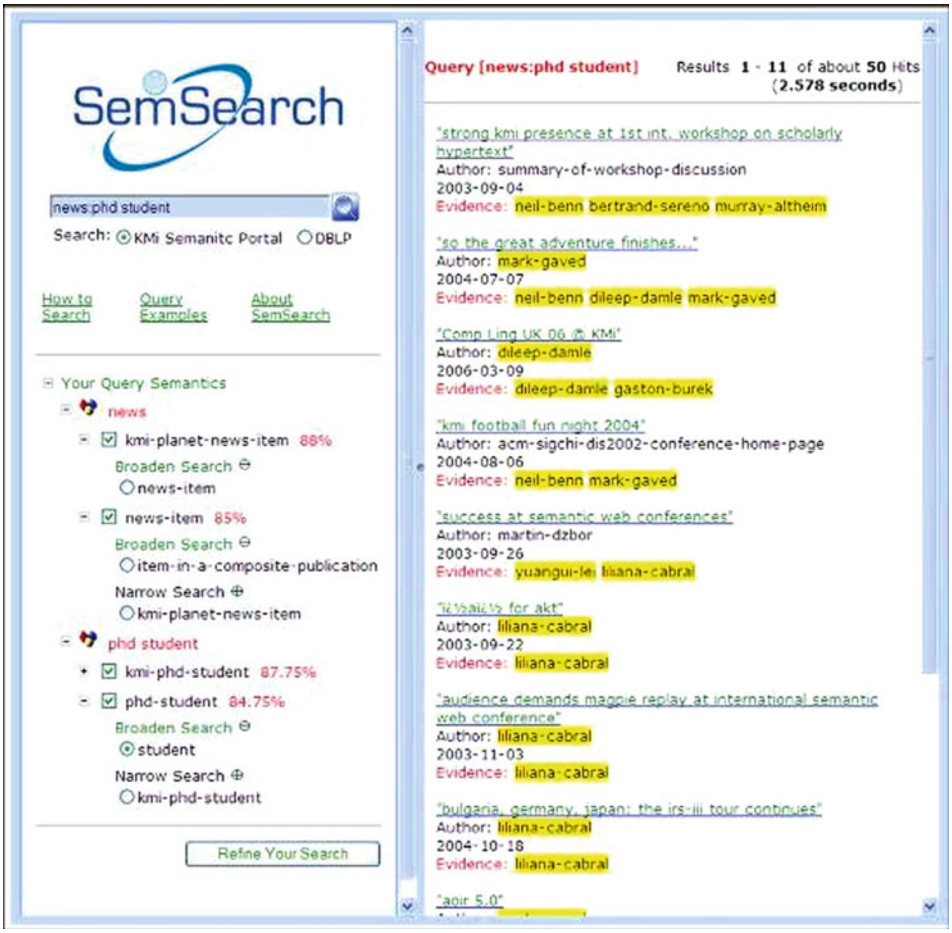
In general, natural language interfaces to ontologies, while potentially useful for naive users, need to be evaluated in practice with large number of users, different ontologies, and large document collections, in order to demonstrate clearly their benefits over the other kinds of retrieval interfaces discussed here.

*Ontology-Based Faceted Browsing*: KIM has a number of front end user interfaces for annotation retrieval and ones customized for specific applications can be easily added.


**Search knowledge about GATE**

| Result: | |
|---|---|
| http://gate.ac.uk//gate/doc/javadoc/gate/creole/class-use/ResourceInstantiationException.html | Source Documentation |
| http://gate.ac.uk//gate/doc/javadoc/gate/creole/class-use/ExecutionException.html | Source Documentation |
| http://gate.ac.uk//sale/tao/split.html | Web Page |
| http://gate.ac.uk//sale/tao/splitch2.html | Web Page |

**Fig. 3.11**
**Language-based interface for semantic annotation retrieval**



news:phd student

Search: [✓ KMi Semanitc Portal] DBLP

Your Query Semantics
- news
  - ☑ kmi-planet-news-item 88%
    - Broaden Search ⊖
      - ○ news-item
  - ☑ news-item 85%
    - Broaden Search ⊖
      - ○ item-in-a-composite-publication
    - Narrow Search ⊕
      - ○ kmi-planet-news-item
- phd student
  - ☑ kmi-phd-student 87.75%
  - ☑ phd-student 84.75%
    - Broaden Search ⊖
      - [✓ student]
    - Narrow Search ⊕
      - ○ kmi-phd-student

Query [news:phd student] Results 1 - 11 of about 50 Hits (2.578 seconds)

'strong kmi presence at 1st int. workshop on scholarly hypertext'
Author: summary-of-workshop-discussion
2003-09-04
Evidence: neil-benn bertrand-sereno murray-altheim

'so the great adventure finishes...'
Author: mark-gaved
2004-07-07
Evidence: neil-benn dileep-damle mark-gaved

'Comp Ling UK 06 @ KMi'
Author: dileep-damle
2006-03-09
Evidence: dileep-damle gaston-burek

'kmi football fun night 2004'
Author: acm-sigchi-dis2002-conference-home-page
2004-08-06
Evidence: neil-benn mark-gaved

'success at semantic web conferences'
Author: martin-dzbor
2003-09-26
Evidence: yuangui-lei liliana-cabral

'[illegible] for akt'
Author: liliana-cabral
2003-09-22
Evidence: liliana-cabral

'audience demands magpie replay at international semantic web conference'
Author: liliana-cabral
2003-11-03
Evidence: liliana-cabral

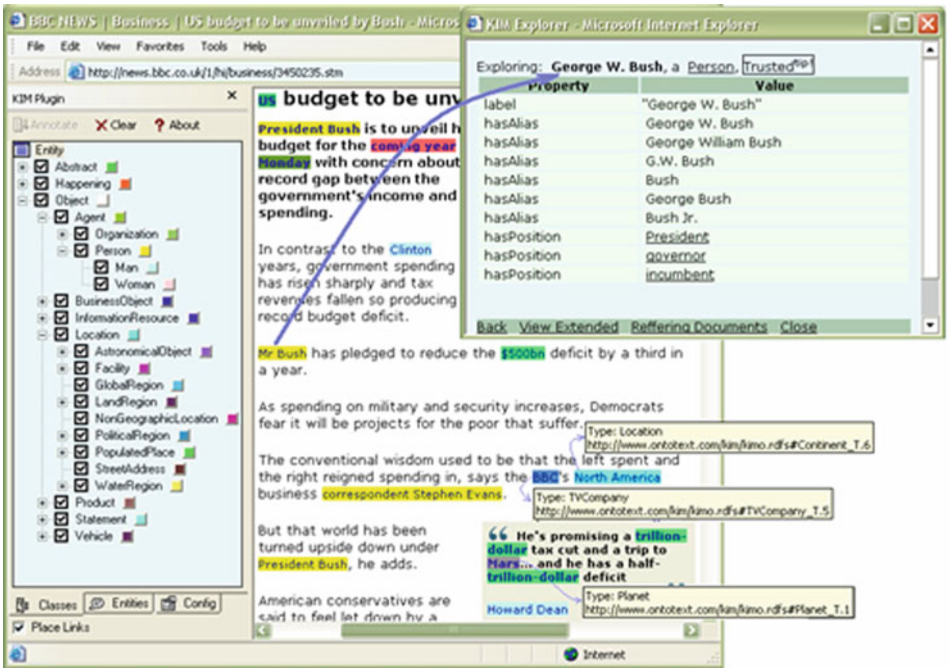
'bulgaria, germany, japan: the irs-iii tour continues'
Author: liliana-cabral
2004-10-18
Evidence: liliana-cabral

'aoir 5.0'

**Fig. 3.12**
**Example SemSearch query results**

The KIM plug-in for Internet Explorer (see *Fig. 3.13*) provides lightweight delivery of semantic annotations to the end user. On its first tab, the plug-in displays the ontology and each class has a color used for highlighting the metadata of this type. Classes of interest are selected by the user via check boxes. The user requests the semantic annotation of the currently viewed page by pressing the Annotate button. The KIM server returns the automatically created metadata with its class and instance identifiers. The results are highlighted in the browser window, and are hyperlinked to the KIM Explorer, which displays further information from the ontology about a given instance (see top right window).

The text boxes on the bottom right in *Fig. 3.13* that contain the type and unique identifier are seen as tooltips when the cursor is positioned over a semantically annotated entity.


BBC NEWS | Business | US budget to be unveiled by Bush - Micros
File Edit View Favorites Tools Help
Address http://news.bbc.co.uk/1/hi/business/3450235.stm
KIM Plugin
Annotate Clear About
Classes Entities Config
Place Links
Internet
KIM Explorer - Microsoft Internet Explorer
Back View Extended Reffering Documents Close

Entity
- Abstract
- Happening
- Object
  - Agent
    - Organization
    - Person
      - Man
      - Woman
  - BusinessObject
  - InformationResource
  - Location
    - AstronomicalObject
    - Facility
    - GlobalRegion
    - LandRegion
    - NonGeographicalLocation
    - PoliticalRegion
    - PopulatedPlace
    - StreetAddress
    - WaterRegion
  - Product
  - Statement
  - Vehicle

US budget to be unv

President Bush is to unveil h budget for the coming year Monday with concern about record gap between the government's income and spending.

In contrast to the Clinton years, government spending has risen sharply and tax revenues fallen so producing record budget deficit.

Mr Bush has pledged to reduce the $500bn deficit by a third in a year.

As spending on military and security increases, Democrats fear it will be projects for the poor that suffer.

The conventional wisdom used to be that the left spent and the right reigned spending in, says the BBC's North America business correspondent Stephen Evans.

But that world has been turned upside down under President Bush, he adds.

American conservatives are said to feel let down by a

"He's promising a trillion-dollar tax cut and a trip to Mars... and he has a half-trillion-dollar deficit

Howard Dean

Type: Location
http://www.ontotext.com/kim/kimo.rdfs#Continent_T.6

Type: TVCompany
http://www.ontotext.com/kim/kimo.rdfs#TVCompany_T.5

Type: Planet
http://www.ontotext.com/kim/kimo.rdfs#Planet_T.1

Exploring: George W. Bush, a Person, Trusted

| Property | Value |
|---|---|
| label | "George W. Bush" |
| hasAlias | George W. Bush |
| hasAlias | George William Bush |
| hasAlias | G.W. Bush |
| hasAlias | Bush |
| hasAlias | George Bush |
| hasAlias | Bush Jr. |
| hasPosition | President |
| hasPosition | governor |
| hasPosition | incumbent |

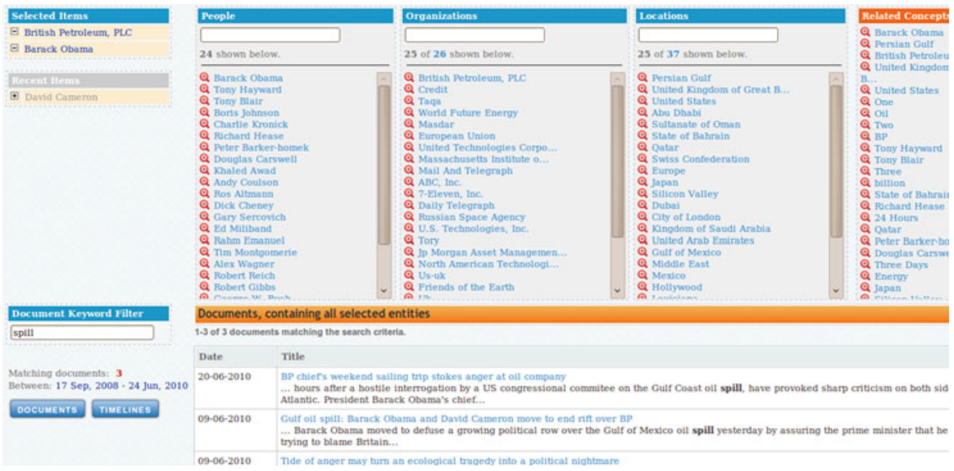
**Fig. 3.13**

**KIM plug-in showing the KIM ontology and KB explorer**

KIM also has a comprehensive Web browser–based UI for semantic search. An important part of that is an annotation retrieval interface, similar to faceted search, where the user can select one or more instances (visualized with their RDF labels, but found via their URIs) and obtain the documents where these are all mentioned.

*Figure 3.14*, for example, shows a case where the user is searching for documents mentioning the entities British Petroleum (BP) and Obama, as well as the keyword spill. As new entities are selected as constraints, the number of matching documents is updated dynamically. At the bottom of the figure, one can see the titles of the retrieved documents and some relevant content from them. The titles can be clicked on in order to view the full document content and the semantic annotations within it. The content of the entity columns (People, Organizations, Locations) is also updated to show only entities contained in the currently retrieved set of documents.

*Mímir* (see http://gate.ac.uk/family/) is a multi-paradigm information management index and repository that can be used to index and search over text, annotations, semantic schemas (ontologies), and semantic metadata (instance data). It allows queries that arbitrarily mix full-text, structural, linguistic, and semantic queries (see *Fig. 3.20*) and that which can scale to gigabytes of text. Its scalability has been tested on semantically annotated data exceeding 10 million documents.



Selected Items
British Petroleum, PLC
Barack Obama

Recent Items
David Cameron

Document Keyword Filter
spill

Matching documents: 3
Between: 17 Sep, 2008 - 24 Jun, 2010

People
24 shown below.
Barack Obama
Tony Hayward
Tony Blair
Boris Johnson
Charlie Kronick
Richard Hease
Peter Barker-homek
Douglas Carswell
Khaled Awad
Andy Coulson
Ros Altmann
Dick Cheney
Gary Sercovich
Ed Miliband
Rahm Emanuel
Tim Montgomerie
Alex Wagner
Robert Reich
Robert Gibbs

Organizations
25 of 26 shown below.
British Petroleum, PLC
Credit
Taqa
World Future Energy
Masdar
European Union
United Technologies Corpo...
Massachusetts Institute o...
Mail And Telegraph
ABC, Inc.
7-Eleven, Inc.
Daily Telegraph
Russian Space Agency
U.S. Technologies, Inc.
Tory
Jp Morgan Asset Managemen...
North American Technologi...
Us-uk
Friends of the Earth

Locations
25 of 37 shown below.
Persian Gulf
United Kingdom of Great B...
United States
Abu Dhabi
Sultanate of Oman
State of Bahrain
Qatar
Swiss Confederation
Europe
Japan
Silicon Valley
Dubai
City of London
Kingdom of Saudi Arabia
United Arab Emirates
Gulf of Mexico
Middle East
Mexico
Hollywood

Related Concepts
Barack Obama
Persian Gulf
British Petroleu
United Kingdom B...
United States
One
Oil
Two
BP
Tony Hayward
Tony Blair
Three
billion
State of Bahrai
Richard Hease
24 Hours
Qatar
Peter Barker-ho
Douglas Carswe
Three Days
Energy
Japan

Documents, containing all selected entities
1-3 of 3 documents matching the search criteria.

| Date | Title |
|---|---|
| 20-06-2010 | BP chief's weekend sailing trip stokes anger at oil company<br>... hours after a hostile interrogation by a US congressional commitee on the Gulf Coast oil **spill**, have provoked sharp criticism on both sid Atlantic. President Barack Obama's chief... |
| 09-06-2010 | Gulf oil spill: Barack Obama and David Cameron move to end rift over BP<br>... Barack Obama moved to defuse a growing political row over the Gulf of Mexico oil **spill** yesterday by assuring the prime minister that he trying to blame Britain... |
| 09-06-2010 | Tide of anger may turn an ecological tragedy into a political nightmare |

**Fig. 3.14**

**KIM's entity-based, faceted annotation retrieval UI**

Since typical semantic annotation projects deal with large quantities of data of different kinds, Mímir provides a framework for implementing indexing and search functionality across all these datatypes, listed below in the order of increasing information density:

*Text*: All semantically annotated documents have a textual content, and consequently, support for full-text search is required in most (if not all) retrieval use cases. Even when semantic annotations are used to abstract away from the actual textual data, the original content still needs to be accessible so that it can be used to provide textual query fragments in the case of more complex conceptual queries.

Mímir uses inverted indexes for indexing the document content (including additional linguistic information, such as Part-Of-Speech or morphological roots), and for associating instances of annotations with the position in the input text where they occur. The inverted index implementation used by Mímir is based on MG4J (http://mg4j.dsi.unimi.it/).

*Semantic Annotations*: The semantic annotation index supports a more generic retrieval paradigm. A unique feature of Mímir is that it can index linguistic, as well as semantic annotations, which thus enables queries mixing the two. For example, if all words in the indexed documents are annotated according to their part of speech and also with semantic classes such as `Person`, `Location`, `Organization`, then the user can pose Mímir queries such as "`CEO of {Mention class==Organization} {Verb}`," which mix text (i.e., CEO of) with semantic and linguistic annotations. A unique and important feature of Mímir is the support for queries nesting one annotation within another, for example, retrieving all semantic annotations of type Person that are contained in document titles. Like many other retrieval systems, Mímir also supports Klene operators.

*Knowledge Base Data*: Knowledge Base (KB) Data consists of an ontology populated with instances. KB data is used to reach a higher level of abstraction over the information in the documents that enables conceptual queries such as finding date ranges or distances. A KB is required for answering such queries because this involves actions like converting from one date format into another and reasoning about scalar values.

A KB that is pre-populated with appropriate world knowledge can perform other generalizations that are natural to human users, such as being able to identify Vienna as a valid answer to queries relating to Austria or Europe.

Mímir uses a Knowledge Base to store some of the information relating to semantic annotations. The links between annotations, the textual data, and the knowledge base information are created by the inclusion into the text indexes of a set of specially created URIs that are associated with annotation data. Furthermore, the URI of entities from the Knowledge Base can be stored as annotation features. The knowledge store used by Mímir for retrieval is based on OWLIM (http://www.ontotext.com/owlim/).

Mímir is discussed and exemplified further in Sect. 3.2.3 below.

## 3.2 Example Applications

### 3.2.1 GATE: A Semantic Annotation Framework

GATE (http://gate.ac.uk, [11]) differs from other manual and automatic semantic annotation systems in that it is a framework. In other words, it provides reusable implementations of semantic annotation components and a set of prefabricated software building blocks that researchers can use, extend, and customize for their specific needs. *Figure 3.15* shows the main semantic annotation components, which will be discussed in more detail next. Conceptually, one can distinguish tools for: (1) manual semantic annotation (i.e., Teamware and OAT in the top half of the figure); (2) document and ontology editing and visualization (the center of the figure); and (3) algorithms for ontology-based information extraction and evaluation (see the lower half).

GATE is implemented in Java and runs on a wide range of platforms. Another distinguishing characteristic of GATE is its *development environment* (called GATE Developer) that helps users minimize the time they spend building new semantic annotation systems or modifying existing ones, by aiding overall development and providing a debugging mechanism for new modules. Because GATE has a plug-in-based model, this allows for the easy coupling and decoupling of the processors, thereby facilitating comparison of alternative configurations of the system or different implementations of the same module (e.g., different parsers). The availability of tools for the easy visualization of data at each point during the development process aids the immediate interpretation of the results.

GATE is engineered to a high standard and supports efficient and robust semantic annotation. It is tested extensively, including regression testing, and frequent performance optimization. GATE has proved capable of processing gigabytes of text and millions of

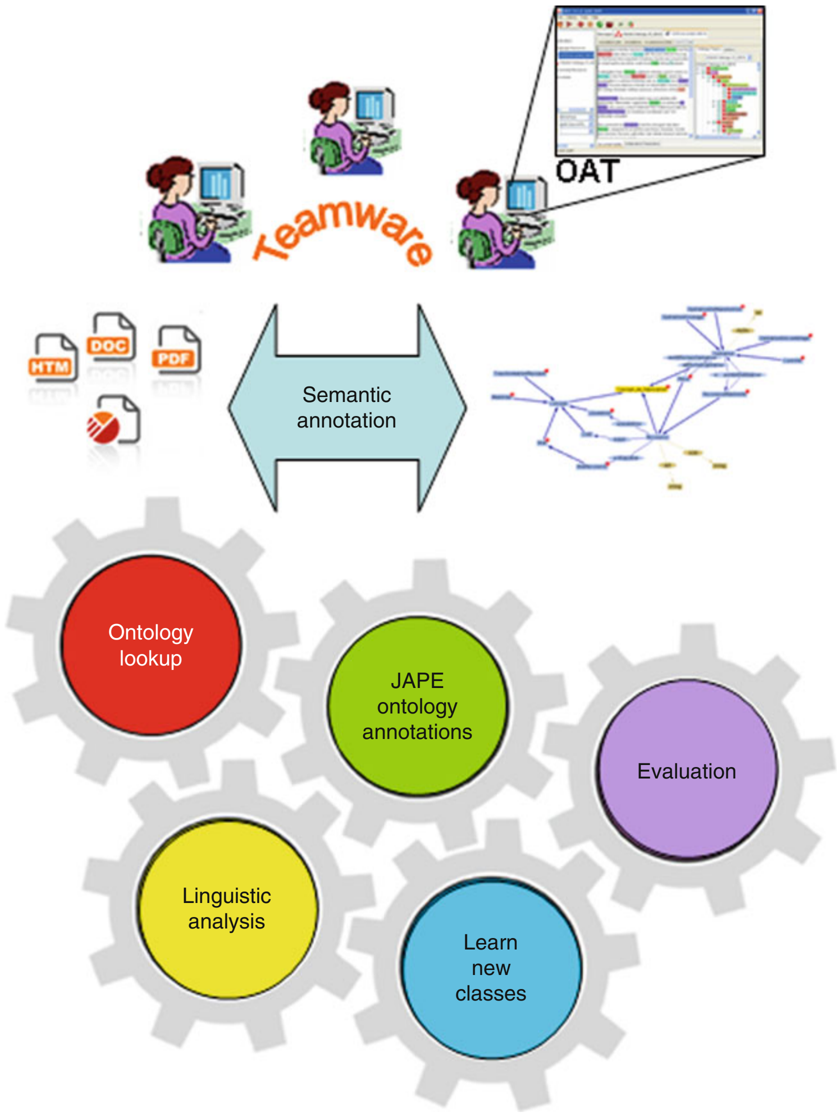


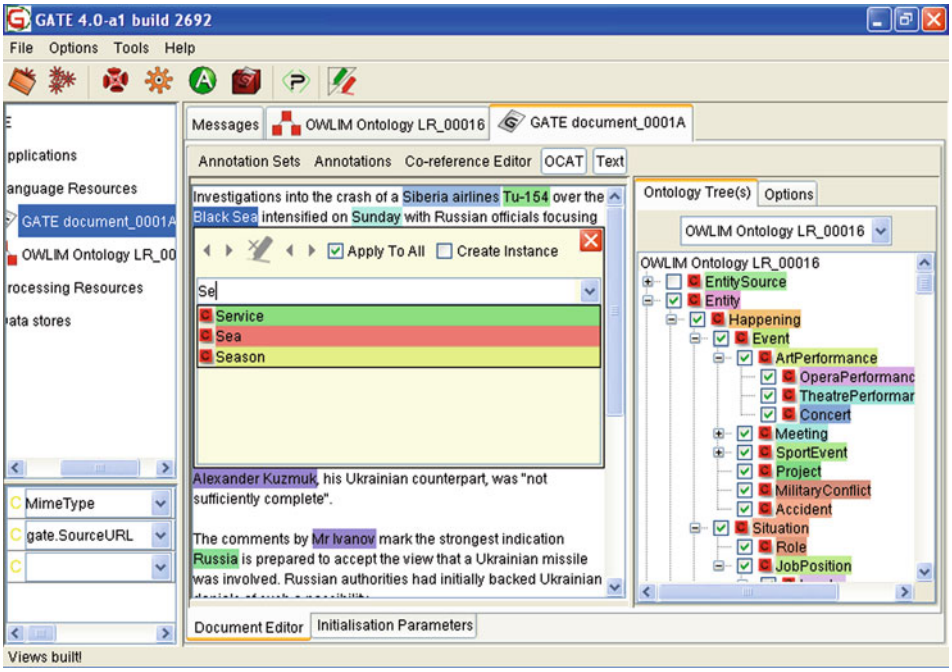
**Fig. 3.15**
**GATE semantic annotation components**

documents. It has been used successfully to build many semantic annotation systems (many discussed in this entry) and ontology learning tools (e.g., Text2Onto [45], Sabou's work [46], SPRAT [47]). The rest of this section discusses the reusable manual annotation tools, the semantic annotation components, and the quantitative evaluation facilities.

First, GATE supports importing, accessing, and visualizing RDF and OWL (see KR and Reasoning on the Semantic Web: OWL) ontologies, as well as using those as lexical and reasoning resources within semantic annotation systems. Since the emphasis is on document annotation, rather than ontology authoring, only basic ontology editing capabilities are provided and the assumption is that typically the ontology would already have been created externally, provided by for example, linked data ontologies. Nevertheless, application-specific extensions with new classes and instances are possible from within GATE.

The second reusable building block is Teamware, which is a collaborative, Web-based annotation tool. It can be used to build both manual and semiautomatic annotation workflows. Alternatively, GATE also provides the single-user, desktop-based Ontology-based Annotation Tool (OAT). *Figure 3.16* shows the process of adding a new annotation, that is, the user highlights part of the text and then starts typing the name of the desired class. A list of possible matches is shown for quick selection. It is possible to create new annotations for all occurrences of the selected text within the given document, thus reducing the manual effort. Similarly, the target instance can be specified or a new instance created, if not already available in the ontology. Once an instance is chosen, it is then possible to annotate property values as well. The editor supports manual annotation with respect to more than one ontology at the same time, by adding the ontology URI to each semantic annotation, in addition to the class and instance URIs.

The most reused and extended components for semantic annotation are the automatic ones, especially the ontology-based gazetteers, the JAPE pattern-matching engine, and the machine-learning facilities.



Investigations into the crash of a Siberia airlines Tu-154 over the Black Sea intensified on Sunday with Russian officials focusing

Alexander Kuzmuk, his Ukrainian counterpart, was "not sufficiently complete".

The comments by Mr Ivanov mark the strongest indication Russia is prepared to accept the view that a Ukrainian missile was involved. Russian authorities had initially backed Ukrainian

**Fig. 3.16**
**OAT: Adding a new annotation**

A gazetteer typically contains names of entities/instances such as cities, organizations, days of the week, etc. The word gazetteer is often used interchangeably for both the set of resources that contain the names and for the algorithm that makes use of those lists to find occurrences of these names in documents. GATE's OntoRoot gazetteer analyzes the ontology, that is, all classes, instances, and properties, to derive a list of lexicalizations (e.g., IBM, Big Blue) and their corresponding URIs (in this example, the URI of the IBM instance). In addition, OntoRoot captures morphological variations, for example, the string "language resources" in the document would be matched against a class with label "language resource." A potential limitation of OntoRoot is that it builds the lexical resources from the ontology only once on initialization, which means that any runtime updates to the ontology are not taken into account as soon as they appear.

The JAPE pattern-matching engine uses rules that describe patterns to be matched (left-hand side) and annotations to be created (right-hand side). It provides access to ontologies on the right-hand side of JAPE rules, which allows rules to add new information to the ontology (e.g., add an instance or a newly discovered property value) or to use reasoning (e.g., to obtain semantic distance between concepts). The ontology and most notably the subsumption relation is also taken into account when matching on the left-hand side. So for example, a rule might look for an organization followed by a location, in order to create a `locatedAt` relationship between them. By using subsumption, the rule automatically matches not just organizations, but also all of its subclasses in the ontology, for example, `Company`, `GovernmentOrg`.

The machine-learning components in GATE provide linguistic information as input to a selection of popular machine-learning algorithms directly from GATE's model of annotations. Once collected, the data are exported in the format required by the ML algorithm, which is often a table where each row is an instance and each column is a feature.

When collecting training data, all the annotations of the type specified as instances are found in the given corpus and for each of them the set of attribute values is determined. All attribute values, provided as features to the learning algorithm, refer either to the current annotation or to one situated at a specified relative position (e.g., +1 is the next annotation). The ML implementation has two modes of functioning: training – when the model is being built, and application – when the built model is used to create new annotations. For an example of how learning can be used for semantic annotation see Sect. 3.1.3, as well as [31].

Another key part of the development of semantic annotation systems is quantitative evaluation. The key metrics applied here are precision, recall, and f-measure.

Precision measures the number of correctly identified items as a percentage of the number of items identified. In other words, it measures how many of the items that the system identified were actually correct, regardless of whether it also failed to retrieve correct items. The higher the precision, the better the system is at ensuring that what is identified is correct.

Recall measures the number of correctly identified items as a percentage of the total number of correct items. In other words, it measures how many of the items that should have been identified actually were identified, regardless of how many spurious identifications were made. The higher the recall rate, the better the system is at not missing correct items.

In general, there is a trade-off between precision and recall, for a system can easily be made to achieve 100% precision by identifying nothing (and so making no mistakes in what it identifies), or 100% recall by identifying everything (and so not missing anything). The F-measure [48] is often used in conjunction with Precision and Recall, as a weighted average of the two.

Since semantic annotation identifies mentions of instances from a given ontology, there are cases when a system would identify an instance successfully but does not assign it the correct class. For example, the entity London in the ontology is an instance of the concept Capital; however, the system annotates the string "London" as belonging to the class City. Since the assigned class does not match the correct class according to the manually annotated data, traditional precision would regard it as wrong. However, due to the closeness of the two classes in the ontology, the system should be given some credit. For such cases, GATE offers BDM (a Balanced Distance Metric), which measures the closeness of two concepts in an ontology or taxonomy [49]. The closer the two concepts are in an ontology, the greater their BDM score is. It is dependent on the length of the shortest path connecting the two classes and also on their depth in the ontology. BDM is normalized with the size of ontology and also takes into account the concept density. In general, BDM can be seen as an improved version of learning accuracy [50].

## 3.2.2 Large-Scale Semantic Annotation of News

Large-scale semantic annotation produces a lot of metadata in the form of annotations. Processing these does not require a heavy reasoning infrastructure, but a scalable infrastructure for Web crawling, automatic annotation, storage, and retrieval. A particular example application to be discussed here is the annotation of news articles, performed by the KIM platform discussed in Sect. 3.1.3 above.

The News Collector demonstrator (http://ln.ontotext.com/) harvests around a thousand articles daily from the Web and has processed over a million news articles since 2002 (At the time of access, not all these articles were available for retrieval in the online demo.). On average, there were around 30 semantic annotations per document and just over 27 million annotations had to be indexed and stored.

In order to achieve the required scalability and real-time semantic annotation, the system has a cluster architecture, shown in *Fig. 3.17*. The cluster provides a set of components that can be configured to work in a distributed environment and allows new processing components to be added on demand. It has centralized repositories for ontologies, semantic annotations, and documents. Scalability of those is achieved through BigOWLIM [51], which is capable of loading and reasoning with over 1 billion of RDF statements. Its performance allows it to replace relational databases in many applications, for example, analytical tasks, business intelligence, and Web front ends to semantic repositories. For an exciting example see BBC's world cup website, which uses BigOWLIM

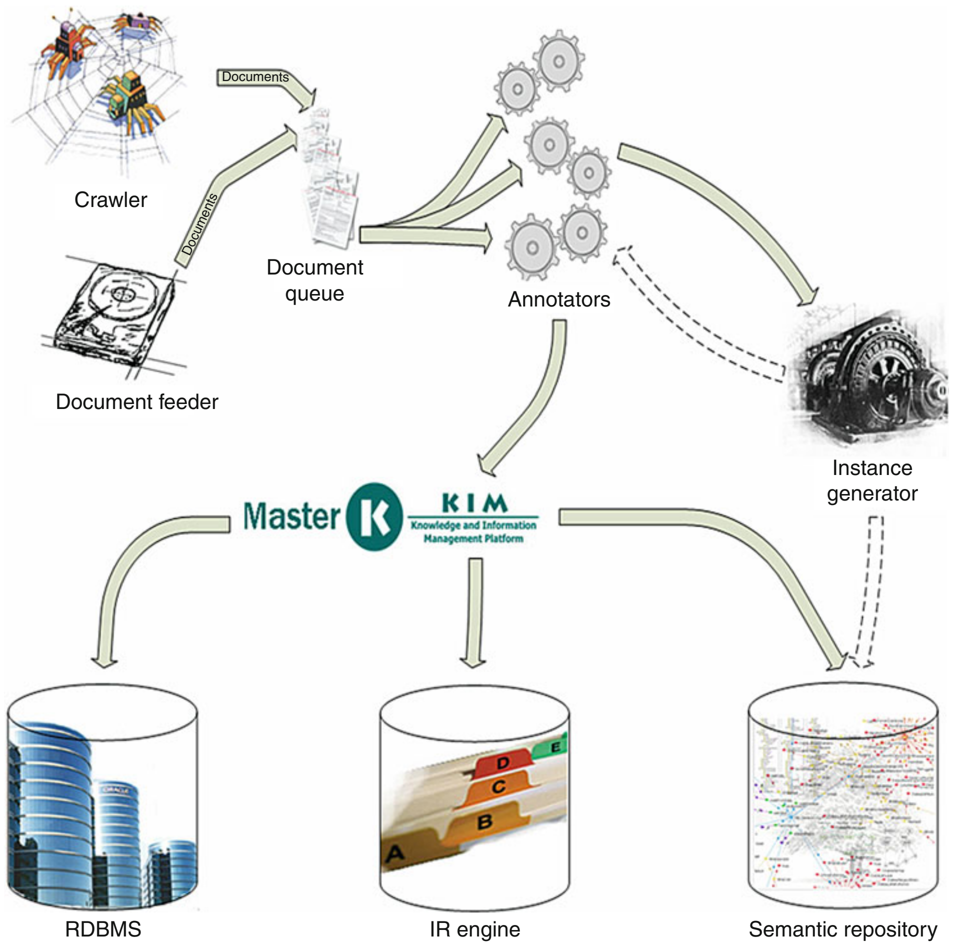


**Fig. 3.17**
**KIM's large-scale cluster architecture**

underneath: http://www.bbc.co.uk/blogs/bbcinternet/2010/07/bbc_world_cup_2010_dynamic_sem.html. The semantic metadata are stored in binary files, which allows instant startup and initialization, because it does not require parsing, re-loading and re-inferring all knowledge, unlike triple-based storage formats.

Annotators are the components of the cluster that have an unlimited number of instances, in order to distribute the computationally heavy semantic annotation task. There is also support for multiple Web crawlers and other data feeders. The cluster supports dynamic reconfiguration, that is, the starting of new crawlers and semantic annotators on demand.

As discussed above, the knowledge bases of large-scale applications, such as News Collector, tend to be billions of RDF triples in size. In contrast, "traditional" information extraction methods typically use much smaller-scale lexical resources, especially gazetteer lists. Therefore, when IE components are adapted to semantic annotation, there is a need for a scalable and fast lookup against the instances in the knowledge base (their labels and properties in particular). Consequently, News Collector has a component called a semantic gazetteer, which runs as one of its annotator modules (others are, e.g., rules for discovery of new instances and relations).

The semantic gazetteer uses the knowledge base to access the entities, their labels, and other properties, as well as some lexical resources (such as possible male person first names). Upon occurrence of a known lexical resource or entity label in the text (e.g., Monday, John, GMT, etc.), the semantic gazetteer generates a semantic annotation with a link to a class in the ontology (e.g., Monday will be linked to the NewsCollector's ontology class DayOfWeek). Moreover, where possible, mentions in the text are linked to the specific instances they refer to (e.g., California will be annotated with the URI of the instance Province.4188).

Since entities can share labels (e.g., New York is both a state and a city), it is often the case that one named entity reference in the text is associated with several possible types and instances. At this stage all possibilities are generated as separate semantic annotations. A subsequent disambiguation component is applied to filter out the irrelevant annotations, based on the text context and other clues.

In addition, News Collector (and KIM in general) distinguishes between prepopulated (or trusted) instances and instances populated automatically during the semantic annotation process. Since the latter can be much less reliable, they are not used by the semantic gazetteer, in order to reduce the propagation of mistakes.

On the other hand, the discovery of new instances or the enrichment of existing instances with new information extracted from the documents is essential in News Collector and other similar systems that deal with very dynamic domains. Due to the size of their knowledge bases, it is not always practical to carry out ontology enrichment manually. Therefore, discovery of such new information becomes a vital part of the semantic annotation process.

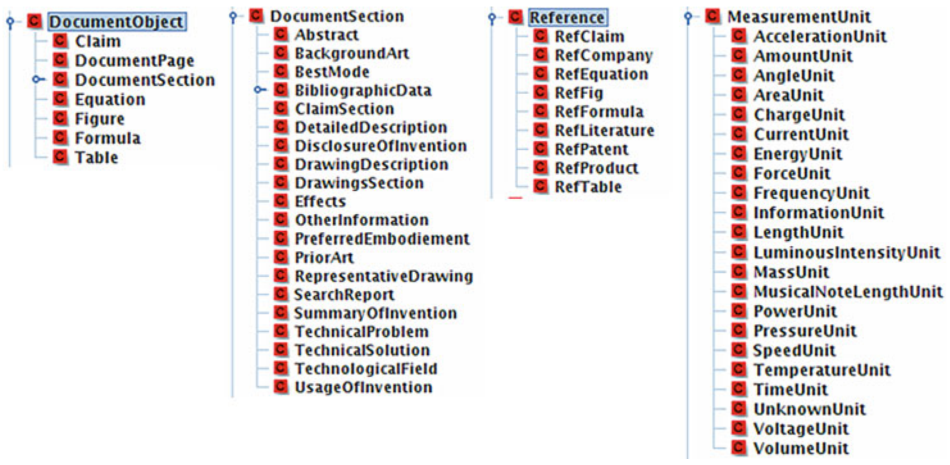
In practical terms, this results in having newly discovered annotations that lack instance information and are thus not linked to the knowledge base via a URI. News Collector uses the IdRF instance disambiguation framework (see Sect. 3.1.4) to either find a matching existing instance and enrich that with the new information, or to create a new instance in the knowledge base. At the end of the semantic annotation process all annotations are linked to the ontology (via their type/class information) and to the knowledge base (via the instance URI). Any relation annotations discovered in the text are used to enrich the KB with new property values (e.g., to assert that David Cameron is UK's prime minister following the May 2010 elections).

## 3.2.3 Large-Scale Semantic Patent Processing

Another large-scale application domain is patent processing. The benefits from semantically enriching patents are threefold. First, semantic annotation is capable of dealing with variable language patterns and format irregularities far easier than text-based regular expressions. For example, references to other patents can be very diverse, for example, US Patent 4,524,128 or Korean laid open utility model application No. 1999-007692. Second, in addition to semantic annotation one can also use an ontology to carry out data normalization. Again, taking an example from references to figures or similarly claims, expressions such as "*Figs. 3.1–3.3*" or "Claims 5–10" imply references not just to the explicitly mentioned figure/claim numbers but also to all those in between. Lastly, automatic semantic annotation techniques are capable of enriching the ever-growing number of patents with more detailed knowledge, which can then be retrieved using multi-paradigm search tools, such as Mímir that combine textual, linguistic, and semantic retrieval.

The SAM project [52] developed an end-to-end, large-scale semantic annotation and retrieval system. One of the main challenges faced in this project is the sheer scale of task. Patent databases typically contain tens of millions of patents, and hundreds of thousands of new ones are produced any year. Worldwide, millions of new patent applications are submitted yearly (see e.g., the statistics page of the World Intellectual Property Organization at http://www.wipo.int/ipstats/). Any application aimed at the IP domain requires a good scalability profile if it is to maintain any credibility.

*Figure 3.18* shows the domain ontology, which models key parts of patent documents, that is, sections, claims, references (e.g., to other patents or publications). Measurements are specific to one or more subject areas and are of interest to specialized patent searchers. Currently, patent professionals use traditional keyword search, but face serious difficulties finding measurements reliably, due to the diverse ways in which they are expressed in language and the need for normalization, for example, some patents have metric units, whereas others use imperial measures. Therefore, measurements are an



- DocumentObject
  - Claim
  - DocumentPage
  - DocumentSection
  - Equation
  - Figure
  - Formula
  - Table
- DocumentSection
  - Abstract
  - BackgroundArt
  - BestMode
  - BibliographicData
  - ClaimSection
  - DetailedDescription
  - DisclosureOfInvention
  - DrawingDescription
  - DrawingsSection
  - Effects
  - OtherInformation
  - PreferredEmbodiement
  - PriorArt
  - RepresentativeDrawing
  - SearchReport
  - SummaryOfInvention
  - TechnicalProblem
  - TechnicalSolution
  - TechnologicalField
  - UsageOfInvention
- Reference
  - RefClaim
  - RefCompany
  - RefEquation
  - RefFig
  - RefFormula
  - RefLiterature
  - RefPatent
  - RefProduct
  - RefTable
- MeasurementUnit
  - AccelerationUnit
  - AmountUnit
  - AngleUnit
  - AreaUnit
  - ChargeUnit
  - CurrentUnit
  - EnergyUnit
  - ForceUnit
  - FrequencyUnit
  - InformationUnit
  - LengthUnit
  - LuminousIntensityUnit
  - MassUnit
  - MusicalNoteLengthUnit
  - PowerUnit
  - PressureUnit
  - SpeedUnit
  - TemperatureUnit
  - TimeUnit
  - UnknownUnit
  - VoltageUnit
  - VolumeUnit

**Fig. 3.18**
**The SAM patent ontology**

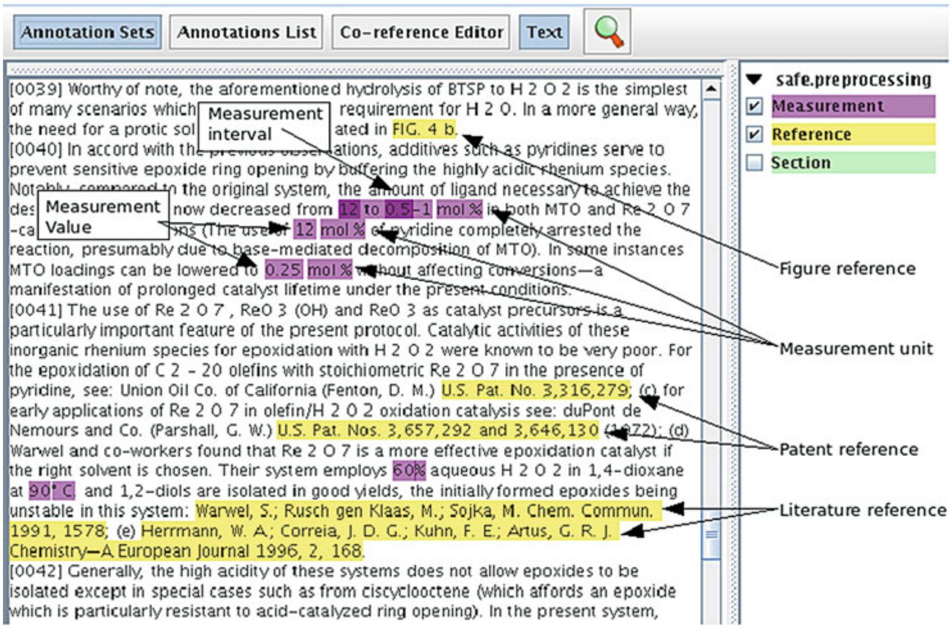
excellent example of the added value and power of automatic semantic annotation and retrieval methods.

The SAM system was developed using GATE [11] and comprises of three types of components: a tokenizer, gazetteer, and a set of semantic annotation rules. These rules are based on patterns and clue words. For example to locate a reference to a table, one rule looks for the clue word table followed by a number. Gazetteers annotate such clue words in the text with all their inflections.

The reference gazetteers are rather small in size, 314 elements in total, and contain clue words such as *Figure*, *Table*, and *Example* to name a few. They also contain entries such as *described in* or *Patent application no.* to help locate literature and patent references.

In the case of measurements, a database (http://www.gnu.org/software/units) containing more than 30 K entries was used to automatically populate a gazetteer list. The database also contains transformation rules for transforming one measurement value into another (e.g., inches to centimeters). Since a gazetteer is simply a list of entries, the information about transforming rules has been populated in the ontology. These rules are used for answering semantic queries by transforming values in one measurement unit into the other on-the-fly.

The application has over 30 rules that identify mentions of the ontology classes in the text (see *Fig. 3.19*). For example, these include identification of complex equations and intervals of measurements. First, the measurement gazetteer is used for identifying measurement units in the text. In the example below, the pattern would annotate text



[0039] Worthy of note, the aforementioned hydrolysis of BTSP to H 2 O 2 is the simplest of many scenarios which requirement for H 2 O. In a more general way, the need for a protic sol ated in FIG. 4 b.
[0040] In accord with th ations, additives such as pyridines serve to prevent sensitive epoxide ring opening by buffering the highly acidic rhenium species. Not the original system, the amount of ligand necessary to achieve the des now decreased from 12 to 0.5–1 mol % in both MTO and Re 2 O 7 -ca ns (The use of 12 mol % of pyridine completely arrested the reaction, presumably due to base-mediated decomposition of MTO). In some instances MTO loadings can be lowered to 0.25 mol % without affecting conversions—a manifestation of prolonged catalyst lifetime under the present conditions.
[0041] The use of Re 2 O 7 , ReO 3 (OH) and ReO 3 as catalyst precursors is a particularly important feature of the present protocol. Catalytic activities of these inorganic rhenium species for epoxidation with H 2 O 2 were known to be very poor. For the epoxidation of C 2 - 20 olefins with stoichiometric Re 2 O 7 in the presence of pyridine, see: Union Oil Co. of California (Fenton, D. M.) U.S. Pat. No. 3,316,279; (c) for early applications of Re 2 O 7 in olefin/H 2 O 2 oxidation catalysis see: duPont de Nemours and Co. (Parshall, G. W.) U.S. Pat. Nos. 3,657,292 and 3,646,130 (1972); (d) Warwel and co-workers found that Re 2 O 7 is a more effective epoxidation catalyst if the right solvent is chosen. Their system employs 60% aqueous H 2 O 2 in 1,4-dioxane at 90° C. and 1,2-diols are isolated in good yields, the initially formed epoxides being unstable in this system: Warwel, S.; Rusch gen Klaas, M.; Sojka, M. Chem. Commun. 1991, 1578; (e) Herrmann, W. A.; Correia, J. D. G.; Kuhn, F. E.; Artus, G. R. J. Chemistry—A European Journal 1996, 2, 168.
[0042] Generally, the high acidity of these systems does not allow epoxides to be isolated except in special cases such as from ciscyclooctene (which affords an epoxide which is particularly resistant to acid-catalyzed ring opening). In the present system,

**Fig. 3.19**

**A semantically annotated patent**

such as “40–50 mph” where 40 and 50 are the two numbers and *mph* is the measurement unit, identified by the gazetteer. As a result, a new annotation of type Measurement will be created, more specifically one of type interval.

```
Rule: MeasurementInterval
(
  {Number}
  {Token.string == "-"}
  {Number}
  {Unit}
):span
-->
:span.Measurement = {type = "interval"}
```

In order to evaluate the consistency in the application’s performance on a large dataset, experiments were carried out on a corpus consisting of 1.3 million US Patent Office documents (108 GB) in XML format with a few attributes on each markup. The average document size was 85 KB. Automatic semantic annotation of all 1.3 million documents took 142 h (5.92 days), at a processing rate of 203.76 KB/s, on a server with 12 threads running in parallel.

In order to be able to estimate the number of semantic annotations that the application produces per document, 20 documents were obtained at random. These contained 147 section annotations, 604 measurements, 1,351 references, and 150,140 linguistic annotations. Based on these results, it would be reasonable to estimate that each document contains an average of 105 semantic annotations and 7,507 linguistic annotations.

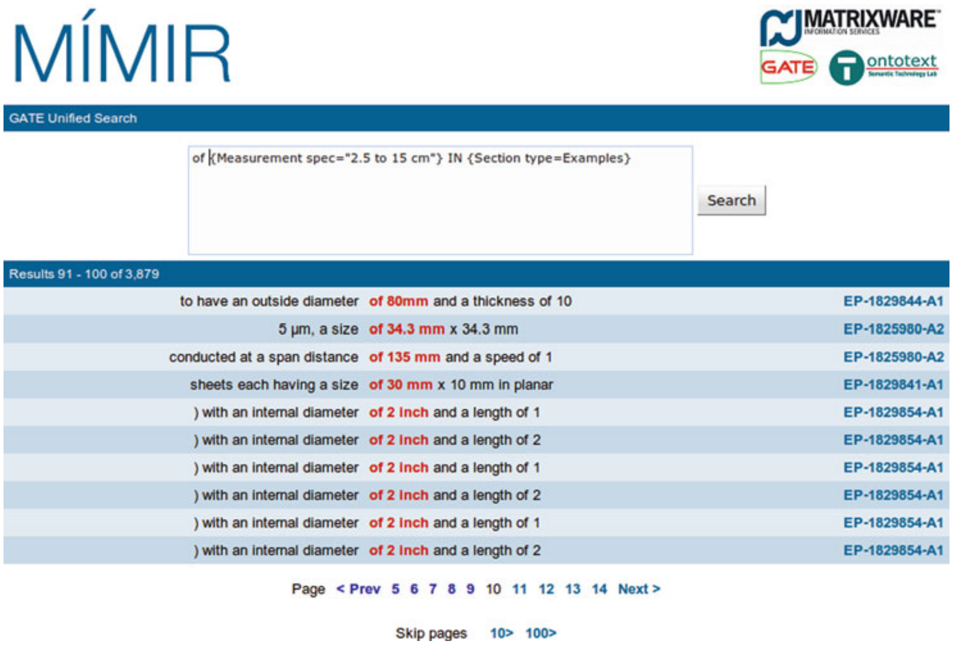
The document content, semantic annotations, and linguistic data were indexed with Mímir, in order to enable semantic annotation retrieval. *Figure 3.20*, for example, shows a multi-paradigm query consisting of the string “of” followed by a measurement semantic annotation with value within the given interval, which must be contained within the examples section. The results show the matching parts of the documents and the surrounding context. As can be seen, the ontology and reasoning has been used to match values like 2 in. or 135 mm to the query range of between 2.5 and 15 cm.

## 3.3 Future Issues

The semantic annotation of Web and intranet content is a research problem that has been receiving significant interest over the past 10 years. As discussed in this entry, automatic and manual approaches have often been combined or used independently, depending on the target application, the volume of the target content, and desired accuracy.

Scalability and large volume, real-time semantic annotation were a challenge until relatively recently, but systems, such as NewsCollector and the patent annotator, have now emerged and are capable of dealing with this challenge.

However, with the emergence of the Web of Data, a new challenge has now emerged. While previously the question was how to annotate millions of documents with



```
of {Measurement spec="2.5 to 15 cm"} IN {Section type=Examples}
```

Results 91 - 100 of 3,879

| | | |
|---|---|---|
| to have an outside diameter | of 80mm and a thickness of 10 | EP-1829844-A1 |
| 5 µm, a size | of 34.3 mm x 34.3 mm | EP-1825980-A2 |
| conducted at a span distance | of 135 mm and a speed of 1 | EP-1825980-A2 |
| sheets each having a size | of 30 mm x 10 mm in planar | EP-1829841-A1 |
| ) with an internal diameter | of 2 inch and a length of 1 | EP-1829854-A1 |
| ) with an internal diameter | of 2 inch and a length of 2 | EP-1829854-A1 |
| ) with an internal diameter | of 2 inch and a length of 1 | EP-1829854-A1 |
| ) with an internal diameter | of 2 inch and a length of 2 | EP-1829854-A1 |
| ) with an internal diameter | of 2 inch and a length of 1 | EP-1829854-A1 |
| ) with an internal diameter | of 2 inch and a length of 2 | EP-1829854-A1 |

Page < Prev 5 6 7 8 9 10 11 12 13 14 Next >

Skip pages 10> 100>

**Fig. 3.20**
**Mímir's annotation retrieval UI**

a small-to-medium-sized ontology, the problem is now how to annotate content with respect to large, interlinked ontologies of billions of RDF triples and millions of instances. Ontologies of this size result in significant ambiguities and thus the onus is now on the instance disambiguation algorithms. However, as discussed in Sect. 3.1.4, further research and especially cross-domain, rigorous evaluations are needed. In addition, with linked data becoming an increasingly important publishing format for analysis results, existing semantic annotation systems need to adapt their conceptual modeling and output mechanisms.

Another considerable open issue is the fact that existing Semantic Web ontologies typically contain very limited linguistic information (i.e., labels), which in turn limits their usefulness as a resource for ontology-based information extraction and semantic annotation. Recent work on linguistically grounded ontologies [53] has recognized this shortcoming and proposed a more expressive model for associating linguistic information to ontology elements. While this is a step in the right direction, nevertheless, further work is still required, especially with respect to building multilingual semantic annotation systems.

Change management and the dynamics of semantic annotations is yet another area that needs to be addressed in future work. The problem arises from the dynamic nature of ontologies (and the world in general). For example, if an ontology is updated with new subclasses or an instance is changed to a class, then the question is what changes need to be made to the semantic annotations and/or the automatic software that created them.

Last but not least, there are some technological challenges, especially the problem of delivering semantic annotation using the Software-as-a-Service model. Unfortunately, content analysis services are currently problematic for both suppliers and customers, for two reasons. First, service creation can have high initial and ongoing infrastructural costs, which are only affordable to very few, large companies as a result. Second, existing semantic annotation services mostly focus on English and a couple of other languages. For example, OpenCalais supports only three languages, is not easily customizable by its users, and involves vendor lock-in. All processed content is also made accessible to the provider (Thomson Reuters in this case), which is not always appropriate due to confidentiality. While there are other proven semantic annotation tools, which are open and easily customizable, they are not yet available as scalable Web Services.

## 3.4 Cross-References

- KR and Reasoning on the Semantic Web: OWL
- Multimedia, Broadcasting, and eCulture
- Semantic Annotation and Retrieval: RDF
- Semantic Annotation and Retrieval: Web of Data

## References

1. McDowell, L.K., Cafarella, M.: Ontology-driven information extraction with OntoSyphon. In: Proceedings of the Fifth International Semantic Web Conference (ISWC 2006), Athens, GA. Lecture Notes in Computer Science, vol. 4273, pp. 428–444. Springer, Berlin (2006)
2. Mahesh, K., Kud, J., Dixon, P.: Oracle at TREC8: a lexical approach. In: Proceedings of the Eighth Text Retrieval Conference (TREC-8), Gaithersburg (1999)
3. Voorhees, E.: Using WordNet for text retrieval. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, pp. 285–303. MIT Press, Cambridge (1998)
4. Handschuh, S., Staab, S.: Authoring and annotation of web pages in CREAM. In: Proceedings of the 11th International World Wide Web Conference (WWW 2002), Honolulu (2002)
5. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: requirements and a survey of the state of the art. J Web Semant. **4**(1), 14–28 (2006)
6. Schroeter, R., Hunter, J.: Annotating relationships between multiple mixed-media digital objects by extending Annotea. In: Proceedings of the Fourth European Semantic Web Conference (ESWC 2007), Innsbruck. Lecture Notes in Computer Science, vol. 4519, pp. 533–548. Springer, Berlin (2007)
7. Halaschek-Wiener, C., Golbeck, J., Schain, A., Grove, M., Parsia, B., Hendler, J.A.: Annotation and provenance tracking in semantic web photo libraries. In: Proceedings of the International Provenance and Annotation Workshop (IPAW 2006), Chicago. Lecture Notes in Computer Science, vol. 4145. Springer, Berlin (2006)
8. Defense Advanced Research Projects Agency: Proceedings of the Sixth Message Understanding Conference (MUC-6). Defense Advanced Research Projects Agency, Morgan Kaufmann, California (1995)

9. Marsh, E., Perzanowski, D.: Muc-7 evaluation of IE technology: overview of results. In: Proceedings of the Seventh Message Understanding Conference (MUC-7), Fairfax. http://www.itl.nist.gov/iaui/894.02/related projects/muc/index.html (1998)
10. ACE: Annotation guidelines for Entity Detection and Tracking (EDT). http://www.ldc.upenn.edu/Projects/ACE/ (2004)
11. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: 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 (ACL 2002), Philadelphia (2002)
12. Kogut, P., Holmes, W.: AeroDAML: applying information extraction to generate DAML annotations from web pages. In: First International Conference on Knowledge Capture (K-CAP 2001), Workshop on Knowledge Markup and Semantic Annotation, Victoria (2001)
13. Fellbaum, C. (ed.): WordNet – An Electronic Lexical Database. MIT Press, Cambridge (1998)
14. Ciravegna, F., Wilks, Y.: Designing adaptive information extraction for the semantic web in Amilcare. In: Handschuh, S., Staab, S. (eds.) Annotation for the Semantic Web. IOS Press, Amsterdam (2003)
15. Maynard, D., Tablan, V., Cunningham, H., Ursu, C., Saggion, H., Bontcheva, K., Wilks, Y.: Architectural elements of language engineering robustness. J. Nat. Lang. Eng. **8**(2/3), 257–274 (2002). Special Issue on Robust Methods in Analysis of Natural Language Data
16. Ciravegna, F.: Adaptive information extraction from text by rule induction and generalisation. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001), Seattle (2001)
17. Ciravegna, F., Dingli, A., Petrelli, D., Wilks, Y.: User-system cooperation in document annotation based on information extraction. In: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002), Siguenza, pp. 122–137 (2002)
18. Motta, E., Vargas-Vera, M., Domingue, J., Lanzoni, M., Stutt, A., Ciravegna, F.: MnM: Ontology driven semi-automatic and automatic support for semantic markup. In: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002), Siguenza, pp. 379–391 (2002)
19. Handschuh, S., Staab, S., Ciravegna, F.: S-CREAM–semi-automatic CREAtion of metadata. In: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002), Siguenza, pp. 358–372 (2002)
20. Handschuh, S., Staab, S., Maedche, A.: CREAM – creating relational metadata with a component-based, ontology-driven framework. In: Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001), Victoria (2001)
21. Baumgartner, R., Froelich, O., Gottlob, G.: The Lixto systems applications in business intelligence and semantic web. In: Proceedings of the Fourth European Semantic Web Conference (ESWC 2007), Innsbruck. Lecture Notes in Computer Science, vol. 4519, pp. 16–26. Springer, Berlin (2007)
22. Domingue, J., Dzbor, M., Motta, E.: Magpie: Supporting browsing and navigation on the semantic web. In: Nunes, N., Rich, C. (eds.) Proceedings of the ACM Conference on Intelligent User Interfaces (IUI 2004), Portugal, pp. 191–197 (2004)
23. Gridinoc, L., Sabou, M., D'Aquin, M., Dzbor, M., Motta, E.: Semantic browsing with PowerMagpie. In: Proceedings of the Fifth European Semantic Web Conference on the Semantic Web (ESWC 2008), Tenerife. Lecture Notes in Computer Science, vol. 5021, pp. 802–806. Springer, Heidelberg (2008)
24. Cimiano, P., Handschuh, S., Staab, S.: Towards the self-annotating web. In: Proceedings of the 13th International World Wide Web Conference (WWW 2004), New York (2004)
25. Shchekotykhin, K.M., Jannach, D., Friedrich, G., Kozeruk, O.: AllRight: automatic ontology instantiation from tabular web documents. In: Proceedings of the Sixth International Semantic Web Conference and Second Asian Semantic Web Conference (ISWC/ASWC 2007), Busan. Lecture Notes in Computer Science, vol. 4825, pp. 466–479. Springer, Berlin (2007)
26. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A., Zien, J.Y.: SemTag

and seeker: boot-strapping the semantic web via automated semantic annotation. In: Proceedings of the 12th International World Wide Web Conference (WWW 2003), Budapest (2003)

27. Mahesh, K., Nirenburg, S., Cowie, J., Farwell, D.: An assessment of Cyc for natural language processing. Technical report MCCS report, New Mexico State University (1966)
28. Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., Goranov, M.: KIM – semantic annotation platform. In: Proceedings of the Second International Semantic Web Conference (ISWC 2003), Sanibel Island. Lecture Notes in Computer Science, vol. 2870, pp. 484–499. Springer, Heidelberg (2003)
29. Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., Kirilov, A., Goranov, M.: Semantic annotation, indexing and retrieval. J. Web Semant. **1**(2), 671–680 (2004). ISWC 2003 Special Issue
30. Mikheev, A., Moens, M., Grover, C.: Named entity recognition without gazetteers. In: Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL 1999), Bergen, pp. 1–8 (1999)
31. Li, Y., Bontcheva, K., Cunningham, H.: Hierarchical, perceptron-like learning for ontology based information extraction. In: Proceedings of the 16th International World Wide Web Conference (WWW 2007), Banff, pp. 777–786 (2007)
32. Gruhl, D., Nagarajan, M., Pieper, J., Robson, C., Sheth, A.: Context and domain knowledge enhanced entity spotting in informal text. In: Proceedings of the Eighth International Semantic Web Conference (ISWC 2009), Chantilly. Lecture Notes in Computer Science, vol. 5823, pp. 260–276. Springer, Berlin (2009)
33. Aswani, N., Bontcheva, K., Cunningham, H.: Mining information for instance unification. In: Proceedings of the Fifth International Semantic Web Conference (ISWC 2006), Athens. Lecture Notes in Computer Science, vol. 4273, pp. 329–342. Springer, Berlin (2006)
34. Fernandez, N., Blazquez, J.M., Sanchez, L., Bernardi, A.: Identityrank: named entity disambiguation in the context of the NEWS project. In: Proceedings of the Fourth European Semantic Web Conference (ESWC 2007), Innsbruck. Lecture Notes in Computer Science, vol. 4519, pp. 604–654. Springer, Heidelberg (2007)
35. Yankova, M., Saggion, H., Cunningham, H.: Adopting ontologies for multisource identity resolution. In: Duke, A., Hepp, M., Bontcheva, K., Vilain, M.B. (eds.) Proceedings of the First International Workshop on Ontology-supported Business Intelligence (OBI 2008), Karlsruhe. ACM International Conference Proceeding Series, vol. 308, p. 6. ACM, New York (2008)
36. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V.C., Sachs, J.: Swoogle: a search and metadata engine for the semantic web. In: Proceedings of the 13th ACM Conference on Information and Knowledge Management (CIKM 2004), Washington, DC (2004)
37. Hildebrand, M., van Ossenbruggen, J., Hardman, J.: /facet: a browser for heterogeneous semantic web repositories. In: Proceedings of the Fifth International Semantic Web Conference (ISWC 2006), Athens, GA. Lecture Notes in Computer Science, vol. 4273, pp. 272–285. Springer, Berlin (2006)
38. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural language interfaces to ontologies: combining syntactic analysis and ontology-based lookup through the user interaction. In: Proceedings of the Seventh Extended Semantic Web Conference (ESWC 2010), Heraklion. Lecture Notes in Computer Science, vol. 6088, pp. 106–120. Springer, Heidelberg (2010)
39. Lopez, V., Uren, V., Motta, E., Pasin, M.: Aqualog: an ontology-driven question answering system for organizational semantic intranets. J. Web Semant. **5**(2), 72–105 (2007)
40. Kaufmann, E., Bernstein, A.: How useful are natural language interfaces to the semantic web for casual end-users? In: Proceedings of the Fourth European Semantic Web Conference (ESWC 2007), Innsbruck. Lecture Notes in Computer Science, vol. 4519. Springer, Berlin (2007)
41. Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., Davis, B., Handschuh, S.: CLOnE: controlled language for ontology editing. In: Proceedings of the Sixth International Semantic Web Conference (ISWC 2007), Busan. Lecture Notes in Computer Science, vol. 4825, pp. 142–155. Springer, Berlin (2007)
42. Bernstein, A., Kaufmann, E.: GINO – a guided input natural language ontology editor. In: Proceedings of the Fifth International Semantic Web

Conference (ISWC 2006), Athens. Lecture Notes in Computer Science, vol. 4273, pp. 144–157. Springer, Berlin (2006)

43. Damljanovic, D., Bontcheva, K.: Enhanced semantic access to software artefacts. In: Proceedings of the Fourth International Workshop on Semantic Web Enabled Software Engineering (SWESE 2008), Karlsruhe (2008)
44. Lei, Y., Uren, V., Motta, E.: Semsearch: a search engine for the semantic web. In: Managing Knowledge in a World of Networks, pp. 238–245. Springer, Berlin/Heidelberg (2006)
45. Cimiano, P., Voelker, J.: Text2Onto – a framework for ontology learning and data-driven change discovery. In: Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB 2005), Alicante (2005)
46. Sabou, M.: From software APIs to web service ontologies: a semi-automatic extraction method. In: Proceedings of the Third International Semantic Web Conference (ISWC 2004), Hiroshima. Lecture Notes in Computer Science, vol. 3298, pp. 410–424. Springer, Berlin (2004)
47. Maynard, D., Funk, A., Peters, W.: Using lexico-syntactic ontology design patterns for ontology creation and population. In: Proceedings of the ISWC Workshop on Ontology Patterns (WOP 2009), Washington, DC (2009)
48. van Rijsbergen, C.: Information Retrieval. Butterworths, London (1979)
49. Maynard, D., Peters, W., Li, Y.: Metrics for evaluation of ontology-based information extraction. In: Proceedings of the Fourth International Workshop on Evaluation of Ontologies for the Web (EON 2006) at the 15th International World Wide Web Conference (WWW 2006), Edinburgh (2006)
50. Cimiano, P., Staab, S., Tane, J.: Automatic acquisition of taxonomies from text: FCA meets NLP. In: Proceedings of the ECML/PKDD Workshop on Adaptive Text Extraction and Mining, Cavtat-Dubrovnik, Croatia, pp. 10–17 (2003)
51. Kiryakov, A.: OWLIM: balancing between scalable repository and light-weight reasoner. In: Proceedings of the 15th International World Wide Web Conference (WWW 2006), Edinburgh (2006)
52. Agatonovic, M., Aswani, N., Bontcheva, K., Cunningham, H., Heitz, T., Li, Y., Roberts, I., Tablan, V.: Large-scale, parallel automatic patent annotation. In: Proceedings of First International CIKM Workshop on Patent Information Retrieval (PaIR 2008), Napa Valley (2008)
53. Buitelaar, P., Cimiano, P., Haase, P., Sintek, M.: Towards linguistically grounded ontologies. In: Proceedings of the Sixth European Semantic Web Conference (ESWC 2009), Heraklion. Lecture Notes in Computer Science, vol. 5554, pp. 111–125. Springer, Heidelberg (2009)