Semantic Search over Documents and Ontologies

Kalina Bontcheva, Valentin Tablan, and Hamish Cunningham

University of Sheffield, Regent Court, 211 Portobello, UK
Initial.Surname@sheffield.ac.uk

1 Introduction

Semantic search over documents is about finding information that is not based just on the presence of words, but also on their meaning [1, 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 word stems as tokens, there has been considerable effort towards using word-senses or lexical concepts (see [3, 4]) for indexing and retrieval. In the case of semantic search, what is being indexed is typically a combination of words, ontological concepts conveying the meaning of some of these words (e.g. Cambridge is a location), and optionally relations between such concepts (e.g. Cambridge is in the UK) [1]. The latter enable somebody searching for documents about the UK to find also documents mentioning Cambridge.

Cambridge however (as well as many other names and words) has multiple meanings, i.e. is ambiguous. The token "Cambridge" may refer to the city of Cambridge in the UK, to Cambridge in Massachusetts, the University of Cambridge, etc. Similarly, different tokens may have the same meaning, e.g. New York and the Big Apple. Therefore, semantic search tries to offer users more precise and relevant results, by using semantics. Frequently this semantics is encoded in ontologies, which can be defined as "a formal specification of a shared conceptualisation" [5]. Alternatively, Google refers to such semantics as *knowledge graphs* and to semantic search as "searching for things, not strings" [6].

Semantic search requires some natural language processing techniques for understanding word meaning. Since some of the most frequently used searches are for persons, locations, organisations, and other named entities [7], one of the most widely used techniques for interpreting this meaning are named entity recognition [8] and semantic annotation [9].

From a retrieval perspective, content annotated with named entities (i.e. PERson, LOCation, etc.) enables semantic search queries such as "LOC earthquake" which would return all documents mentioning a location and the word earthquake. Semantic annotation, on the other hand, goes one step further by differentiating, among other things, which specific real-world location is mentioned in the text (e.g. Cambridge, UK vs Cambridge, Mass.). This enables even more

N. Ferro (Ed.): PROMISE Winter School 2013, LNCS 8173, pp. 31–53, 2014.

powerful searches for documents, based on knowledge and relationships that are external to those documents. For example, a query on flooding in the UK would retrieve a document about floods in Sheffield, even if the latter does not explicitly mention the UK.

As can be seen from these example, what is required here is a source of knowledge that there are several cities called "Cambridge", one in the UK (which is a country) and one in Massachusetts, which is a state in the USA. Semantic annotation typically uses ontologies which contain such semantic knowledge, not just about the concepts (e.g. country, city) and instances (e.g. UK, USA) but also about relationships between concepts (e.g. cities are located within countries) and relations between instances (e.g. Cambridge_UK is located within England, which in turn is located within the UK).

Some semantic annotation methods have used Wikipedia as the large-scale source of such knowledge, e.g. [10–12]. However, recent semantic annotation and search methods have increasingly turned towards exploiting the massive, interlinked cloud of Linked Open Data (LOD)[1], which contains hundreds of billions of statements about entities and relations between them. Some LOD datasets cover general knowledge (e.g. DBpedia (automatically derived from Wikipedia), YAGO, Freebase), whereas others focus on domain-specific knowledge (e.g. MusicBrainz, PubMed, GeoNames). Some state-of-the-art methods for semantic annotation (and consequently search) utilise only one LOD resource, (e.g. DBpedia Spotlight[13]), YAGO (e.g. [14]), MusicBrainz (e.g. [15]), whereas others create a larger knowledge resource, mixing several different LOD datasets. For instance, LDSR is a collection of several LOD datasets, comprising 440 million explicit statements about entities, derived from DBpedia, Geonames, Wordnet, the CIA Factbook, lingvoj, and UMBEL [16]. Similarly, Google's knowledge graph contains 500 million entities and is derived from Wikipedia, the CIA Factbook, and Freebase [6].

The rest of this chapter is structured as follows. Section 2 introduces semantic search in more detail, followed by a discussion on why semantic search is necessary (Section 3). Next Section 4 discusses where does document semantics comes from and how it is indexed. Semantic search query languages and available semantic search engines are discussed in Section 5. Due to the complexity of the underlying query languages, user-friendly interfaces to semantic search are key (see Section 6). Next Section 7 discusses evaluation, followed by a conclusion, which outlines outstanding challenges in this area.

2 What Is Semantic Search

In order to understand better what semantic search is, it is useful to consider two aspects: (i) what is being searched; and (ii) what are the results. We discuss these in turn next.

With respect to what is being searched, there are three main kinds of content to consider:

[1] `http://linkeddata.org/`

- *Documents*: This is traditional full-text search, where queries are answered on the basis of word co-occurrence in text content. For example, a query for "Cambridge university" returns all documents that contain the words Cambridge and/or university somewhere. This does not mean the results are only documents about that university. This kind of search has problems answering entity-type queries, e.g. which cities in the UK have population less than 100,000.
- *Ontologies and other semantic knowledge, e.g. LOD*: This is search over structured formal data, typically expressed as RDF [17] or OWL [18], and stored in a database or a semantic repository. Consequently, such formal queries are expressed in structured query languages, such as SPARQL [19] or SQL. This kind of search is often referred to as semantic search, because it uses semantics and inference to find the matching formal knowledge. In this chapter, we will refer to this kind of search as *ontology-based search*. This kind of search is particularly suited to answering entity-type queries, such as our example above.
- *Both documents and formal knowledge*: This is what this chapter refers to as semantic search over documents, or multi-paradigm [2], or semantic full-text search [20]. This kind of search draws both on document content and on semantic knowledge, in order to answer queries such as: "flooding in cities in the UK" or "flooding in places within 50 miles of Sheffield". In this case information about which cities are in the UK or within 50 miles of Sheffield is the result of ontology-based search (e.g. against DBpedia or GeoNames). Documents are then searched for the co-occurrence of the word "flooding" and the matching entities from the ontology-based search. In other words, what is being searched here are the document content for keywords, the index of semantically annotated entities that occur within these documents, and the formal knowledge.

With respect to the kinds of results returned by searches, there are four main kinds:

- *Documents*: The search returns a ranked list of documents, typically displayed with their title and optionally, some additional metadata (e.g. author). This kind of results are typically produced by full-text searches, although some do also include snippets.
- *Documents + highlighted snippets*: In addition to document titles, one or more snippets are returned, where the query hits are highlighted, in an attempt to make it apparent to users why this document is relevant to their query. Semantic search systems typically return matching documents in this way, e.g. the KIM system [1], Mímir[2], Broccoli [21].
- *Information summary*: This is a human-readable rendering of formal knowledge, returned by ontology-based searches for entities. For instance, a search in Google for "Tony Blair" would display on the right a summary showing several photos and basic facts, such as date of birth, generated automatically from their formal knowledge graph representation [6].

– *Structured results*: Ontology-based searches, which results are a list of entities, are often shown like that, e.g. a list of UK city names. See for example the KIM entity searches[2] [1] or Brocolli [21].

3 Why Semantic Full-Text Search

As argued by [20], full-text search works well for precision-oriented searches, when the relevant documents contain the keywords that describe the user need. However, there are many cases when recall is paramount and, also, implicit knowledge is needed in order to answer parts of the query. A frequent class of such queries are entity-based ones, e.g. "plants with edible leaves" [20]. In this case, most likely there is no one document containing the answer, and also, documents most likely refer to the specific plants by name (e.g. broccoli), instead of using the generic term "plants".

Environmental science is another example, where there is a strong need to go beyond keyword-based search [22]. The British Library carried out a survey of environmental science researchers and analysed the kinds of information needs they struggled to satisfy through keyword search [23]. The top requirement was for geographically specific queries, including proximity search (e.g. "documents about flooding within 50 miles of Sheffield") and implied locations (e.g. the query "documents about flooding in South West England" needs to return a document about flooding in Exeter, even though South West England is not mentioned explicitly).

One more example is patent search [24], where recall is crucial, since failure to find pre-existing, relevant patents may result in legal proceedings and financial losses. Examples of hard to find information using keywords alone are searches for references to papers cited in a specific section of the patent and also searches for measurements and quantities (e.g. in chemical patents). Measurements in particular are numeric and can show great variation – the same value can be expressed using different measurement systems, e.g. inches or centimetres, or different multipliers even when using the same measurement system, e.g. mm, cm, or metres.

Implementing semantic full-text search poses four key challenges [20]:

1. The automatic recognition of entities in full text, since most content is not pre-annotated.
2. Indexing efficiently words, entity occurrences, and formal knowledge in ontologies.
3. Need for semantic interpretation of document content and search query.
4. Easy to use and transparent user interfaces for semantic full-text search.

Next we discuss the first three challenges in more detail, while the last challenge is covered in Section 6.

[2] `http://ln.ontotext.com/KIM`

4 Indexing Document Semantics

As can be seen from our discussion above, understanding the meaning of documents is the key enabler of semantic search. Two main approaches have been explored so far: (i) asking document creators to encode semantics in a machine-readable format at publishing time; (ii) deriving semantics automatically.

The advantage of the first approach is that it relies on human intelligence and could thus be more accurate. The disadvantages are that humans need to learn and adopt these metadata encoding formats and do that for a sufficiently large number of documents. Lastly, already published content would need to be tagged retrospectively, which is not always possible.

Conversely, the advantage of automatic approaches is that machine readable semantics can be generated for every document, even already published ones, regardless of their publication format. The disadvantage is loss of accuracy, since semantic annotation methods tend to perform with between 80 and 90% precision and recall.

Next we discuss in some detail each of the two approaches.

4.1 Human-Encoded Semantics

Human-embedded semantics in web documents typically conforms to one of two standards: RDFa (http://rdfa.info/) and Schema.org.

In brief, RDFa (or Resource Description Framework in attributes) is a W3C standard. It adds a set of attribute-level extensions to HTML and XHTML, to describe rich metadata, embedded within Web documents.

Schema.org is a similar endeavour, which is supported by many of the web search engines, including Bing, Google, Yahoo, and Yandex.

This additional metadata has enabled search engines to enrich their result presentation, by adding richer information, e.g. a restaurant's phone number and address [25]. Even though useful for some queries, such metadata nevertheless falls short of enabling the kinds of semantic queries discussed above.

4.2 Automatic Semantic Annotation

The process of tying semantic models and natural language together is referred to as *semantic annotation*. It may be characterised as the dynamic creation of interrelationships between *ontologies* and unstructured and semi-structured documents in a bidirectional manner [1]. 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.

Figure 1 shows an example text on the left. The automatically added semantic annotation on the phrase "South Gloucestershire" is shown on the right. As can seen, the entity mention has now been linked to the corresponding DBpedia (the value of the inst attribute) and GeoNames URIs (the value of the geonamesURI attribute). Additional semantic knowledge has been brought in from these LOD

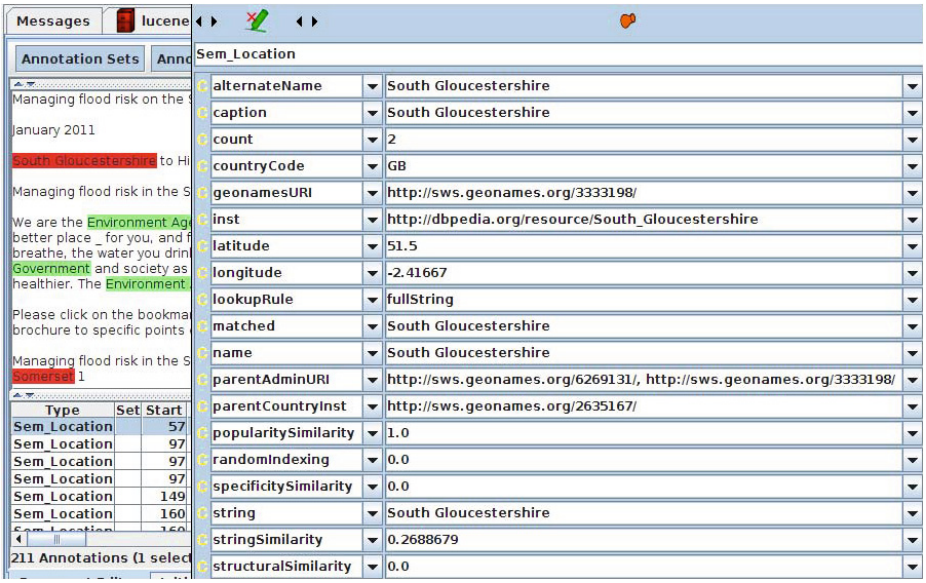

Fig. 1. Example Semantic Annotation in GATE [26]

resources, including latitude and longitude, the URI of the parent country, and the URIs of the relevant NUTS administrative regions.

Information Extraction (IE), a form of natural language analysis, has become the natural language processing technology of choice, for bridging the gap between unstructured text and formal knowledge expressed in ontologies. Ontology-Based IE (OBIE) is IE which is adapted specifically for the semantic annotation task [27]. 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. [28]) call ontology-based any system which specifies its outputs with respect to an ontology, however, in our view, 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 type of the extracted entity (i.e. Location for South Gloucestershire), but it also disambiguates it, by linking it to its semantic description in the target knowledge base, typically via a URI (see Figure 1). This allows entities to be traced across documents and their descriptions to be enriched during the IE process, as shown. In practical terms, this requires automatic recognition and disambiguation of named entities, terms, and relations and also co-reference resolution both within and across documents. These more complex algorithms are typically preceded by some shallow linguistic pre-processing (tokenisation, Part-Of-Speech (POS) tagging, etc.) In our example, entity recognition and disambiguation have been carried out by the open-source GATE natural language toolkit [26, 29].

Linking Open Data resources, especially DBpedia, YAGO and Freebase, have become key sources of ontological knowledge for semantic annotation, as well as being used as target entity knowledge bases for disambiguation. These offer: (i) cross-referenced domain-independent hierarchies with thousands of classes and relations and millions of instances; (ii) an inter-linked and complementary set of resources with synonymous lexicalisations; (iii) grounding of their concepts and instances in Wikipedia entries and other external data. The rich class hierarchies are used for fine-grained classification of named entities, while the knowledge about millions of instances and their links to Wikipedia entries are used as features in the OBIE algorithms.

Due to space limitations, we are not able to discuss semantic annotation methods and approaches in more details, but see [30] for details.

5 Semantic Search Approaches: A High-level Overview

As discussed already, semantic annotations 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 semantic 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 be very substantial, as the cardinality of the annotation sets grows. Therefore different, more optimised solutions have been investigated.

The main difference from semantic web search engines, such as Swoogle [31], 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 [32], 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.

Before discussing several representative semantic search approaches in more detail, let us first discuss query languages for semantic search.

5.1 Semantic Search Queries

Since semantic search queries need to contain both text-based keywords and formal SPARQL-like queries over the ontology, they are often referred to as

hybrid queries. The Semplore system [33], for example, uses conjunctive hybrid query graphs, similar to SPARQL, but enhanced with a "virtual" concept called keyword concept W. A similar approach has been taken in the Broccoli system [21], which has a special $occurs - with$ relation, the value of which is the free text keyword.

Mímir[3] [2] has an even richer query language, which also supports the inclusion of linguistic annotations in queries. For example, a Mímir query "PER says" will return documents where an entity of type Person is followed by the keyword says. Morphological variations for keywords are also supported (e.g. "PER root:say"), as are distance restrictions (e.g. "Person [0..5] root:say" which matches text such as "Sebastian James of Dixons Group said"). Additional semantic restrictions based on knowledge from the ontology are expressed by adding a SPARQL query. For examples, this query is for documents mentioning people born in Sheffield:

```
{Person sparql = "SELECT ?inst
    WHERE { ?inst :birthPlace <http://dbpedia.org/resource/Sheffield>}"}
```

5.2 Relevance Scoring and Retrieval

In the context of semantic full-text search, [34] propose a modification of TF.IDF, based on the frequency of occurrence of instances from the semantic annotations in the document collection. They also combine semantic similarity with a standard keyword-based similarity for ranking, in order to cater for cases when there are no sufficiently relevant semantic annotations.

The Mímir semantic full-text search framework (see Section 5.3) supports different ranking functions and new ones can easily be integrated. In addition to TF.IDF, it already implements ranking based on hit length and the BM25 algorithm.

The CE[2] system goes one step further and uses a graph-based approach to compute the ranking of the hybrid search results [35]. The graph structure comes from the formal semantic knowledge.

With respect to ranking individuals returned via knowledge base search, [36] propose ObjectRank – a PageRank-based approach.

5.3 Semantic Search Full-Text Platforms

Due to limited space, the focus of this section is on extensible semantic search full-text frameworks (namely KIM and Mímir), which are available to download and experiment with freely for research. There is other relevant research, mostly at the level of proof-of-concept prototypes, some of which are introduced briefly next.

GoNTogle [37] is a search system that provides keyword, semantic and hybrid search over semantically annotated documents. The semantic search replaces

[3] A set of example queries and several test Mímir indexes are available for experimentation at: `http://demos.gate.ac.uk/mimir/`

keywords with ontological classes. Results are obtained based on occurrences of the ontological classes from the query within the annotations associated with a document. Finally, the *hybrid search*, comprises a standard boolean AND or OR operation between the result sets produces by a keyword search and a semantic search. The only type of annotation supported is associating an ontology class with a document segment. Another similar system is Semplore [33] which uses conjunctive hybrid query graphs, similar to SPARQL, but enhanced with a "virtual" concept called keyword concept W. However, both GoNTogle and Semplore do not have support for searches over document structure, nor for searches over other types of linguistic annotations.

The Broccoli system [21] also provides a user interface for building queries, combining text-based and semantic constraints (encoded as entity mentions in the input text, with URIs). The association between text and semantics is encoded by means of the *occurs-with* relation which is implied whenever mentions of words and ontological entities occur within the same *context*. The *contexts* are automatically extracted at indexing time, and rely mainly on shallow syntactic analysis of the document and extraction of syntactic dependency relations. The *occurs-with* relation provides access to the underlying phrase structure of the input document. However, the system is designed to only use this particular relation, so indexing other kinds of document structure (e.g. abstract, sections) is likely to prove problematic. Consequently, there is no support for richer linguistic annotations, such as part-of-speech or morphology, document metadata, or structural search other than based on co-occurrences within *contexts*.

The KIM. (Knowledge and Information Management) platform [1, 38], was among the first systems to implement semantic search, both over RDF knowledge bases via SPARQL, and over semantically annotated document content, including hybrid queries mixing keywords and semantic restrictions. KIM has a number of user interfaces for semantic search and browsing and can be customized easily for specific applications. It is freely available for research use from http://www.ontotext.com/kim/getting-started/download.

KIM is an extendible platform for knowledge management, which offers tools for semantic annotation, indexing, and semantic-based search (referred to as multi-paradigm search in KIM). Figure 2 shows KIM's architecture, which also includes a web crawler for content harvesting; a knowledge ETL component which interfaces to thesauri, dictionaries, and LOD resources; and a set of web-based user interfaces for entity-based and semantic-based full text search (see 6.1 for details on the KIM faceted search).

Semantic annotation in KIM is based on the open-source GATE NLP framework [29]. 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.

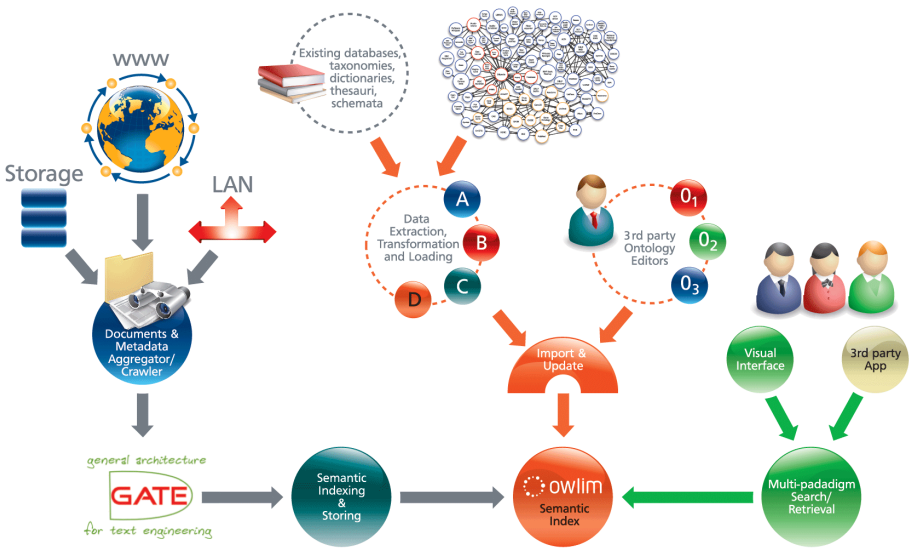

Fig. 2. KIM Architecture

KIM can also use Linked Data ontologies for semantic annotation and search. At present it has been tested with DBPedia, Geonames, Wordnet, Musicbrainz, Freebase, UMBEL, Lingvoj and the CIA World Factbook. Those datasets are preprocessed and loaded to form an integrated dataset of about 1.2 billion explicit statements. Forward-chaining is performed to materialise another 0.8 billion implicit statements.

Mímir [4] [2] is an integrated semantic search framework, which offers indexing and search over full text, document structure, document metadata, linguistic annotations, and any linked, external semantic knowledge bases. It supports hybrid queries that arbitrarily mix full-text, structural, linguistic and semantic constraints. A key distinguishing feature from previous work are the containment operators, that allow flexible creation and nesting of full-text, structural, and semantic constraints.

Figure 3 shows the Mímir semantic query UI. In this case the goal is to find documents, mentioning locations in the UK, where the population density is more than 500 people per square km. In this case the knowledge about population density is coming from DBpedia. The documents being searched in this case are metadata descriptions of government reports on climate change and flooding, created by the British Library as part of the EnviLOD project[5].

The high-level concept behind Mímir is that a document collection is processed with NLP algorithms, typically including semantic annotation using Linked

[4] http://gate.ac.uk/mimir/
[5] http://gate.ac.uk/projects/envilod

Fig. 3. Mímir's Semantic Search UI showing a formal query, the retrieved documents, and short text snippets showing in bold the matched locations

Open Data accessed via a triple store, such as OWLIM [39] or Sesame. The annotated documents are then indexed in Mímir, together with their full-text content, document metadata, and document structure markup (the latter can also be discovered automatically via the NLP tools). At search time, the triple store is used as a source of implicit knowledge, to help answer the hybrid searches that combine full-text, structural, and semantic constraints. The latter are formulated using a SPARQL query, executed against the triple store.

Mímir uses inverted indexes for indexing the document content (including additional linguistic information, such as part-of-speech or morphological roots), and for associating instance of annotations with the position in the input text where they occur. The inverted index implementation used by Mímir is based on MG4J [40]. Beside document text, the other main kind of data are the structural and NLP-generated annotations. In Mímir both kinds are represented in the same data structure, comprising a start and end position, an annotation type (e.g. Location, p), and an optional set of attributes (called features in the GATE framework).

Mímir is highly scalable. 150 million web pages were indexed successfully, using two hundred Amazon EC2 Large Instances running for a week to produce a federated index [41]. Since Mímir runs on GateCloud.net [42], building Mímir semantic indexes on the Amazon cloud is straightforward.

6 User Friendly Semantic Search Interfaces

SPARQL-like semantic searches that tapping into ontological knowledge and LOD resources are extremely powerful. However, writing formal SPARQL queries is beyond the ability of the vast majority of users.

This section discusses a number of user-friendly interfaces for semantic search, which we have classified in the following three categories:

- Ontology-based faceted search interfaces;
- Form-based search interfaces;
- Text-based searches (natural language interfaces for semantic search).

6.1 Ontology-Based Faceted Search

As discussed earlier, KIM has a comprehensive set of Web browser-based UIs for semantic search. One particular kind is ontology-driven faceted search, where the user can select one or more instances (visualised with their RDF labels, but found via their URIs) and obtain the documents where these co-occur. Timeline and entity-centric views are also supported.

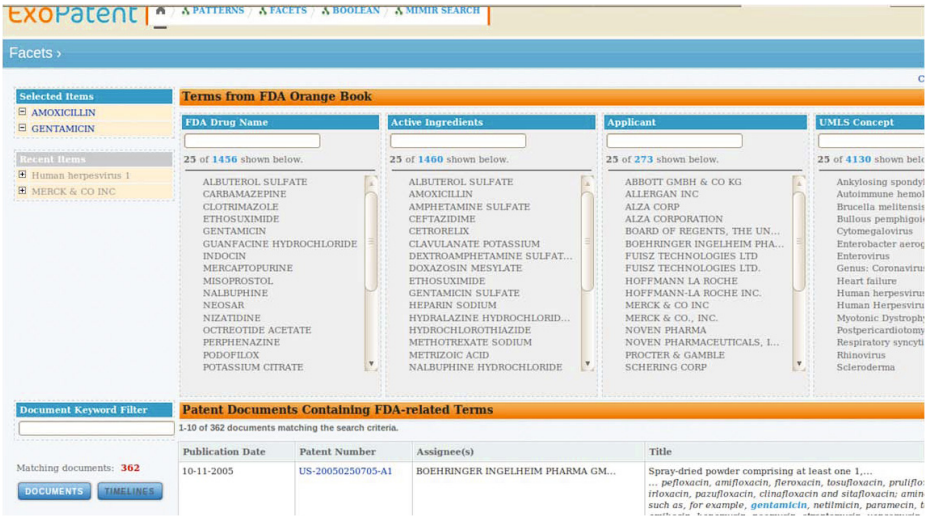

Fig. 4. KIM's entity-based faceted search UI

Figure 4, for example, shows a case where the user is searching for patents mentioning amoxicillin and gentamicin. This example is taken from the ExoPatent online KIM demo[6], which uses the the FDA Orange Book (23,000 patented drugs) and Unified Medical Language System (UMLS – a database

[6] Available online at `http://exopatent.ontotext.com`

of 370,000 medical terms) to annotate documents with semantic information. The demo runs on a small set of 40,000 patents. ExoPatent supports semantic search for diseases, drug names, body parts, references to literature and other patents, numeric values, and ranges.

In the faceted search UI, as new entities are selected as constraints (see left column), the number of matching documents is updated dynamically. Optional keyword constraints can also be specified in the keyword filter field on the left. At the bottom of the figure, one can see the titles of the retrieved documents and some relevant content from them. The titles are clickable, in order to view the full document content and the semantic annotations within it. The entities/terms listed in the entity columns (i.e. Drug name, ingredients, applicant, and UMLS concept) are also updated, to show only entities co-occurring with the already selected entity constraints.

The Broccoli system [21] has a similar interactive query building UI, which updates dynamically as the user is typing concepts or keywords to search for. What is being searched are Wikipedia articles, indexed with classes and instances from the YAGO ontology. Figure 5 shows an example query for documents mentioning UK cities, which also contain the keyword "flood". The semantic query is displayed as a graph on top, making explicit the relations between the searched for concepts. Keywords have a special relation "occurs-with", whereas all other semantic relations come from the YAGO ontology. As the user starts typing a query term (e.g. City), the lists of matching classes, instances, and relations on the left are updated dynamically. For instance, once City is selected, only relations applicable to this class are shown in the list of relation candidates. Due to the entity-centric queries, the result list is structured as a list of entities, where relevant information from the YAGO ontology is provided for each returned entity, as well as documents from Wikipedia about this entity, which also contain the given keyword(s).

6.2 Natural Language Queries

NL queries allow users to perform semantic search through written language (e.g. English). In general, natural language search interfaces, while potentially useful for naive users, are still fairly experimental. The majority of work on natural language interfaces for semantic search has focused on the problem of querying ontologies (e.g. [43–45]) or ontology authoring (e.g. [46, 47]). 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 [48], for example, has an ontology modelling documents 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 artefacts (software code, documentation, user manuals, papers, etc.) were annotated semantically with respect to a domain ontology.

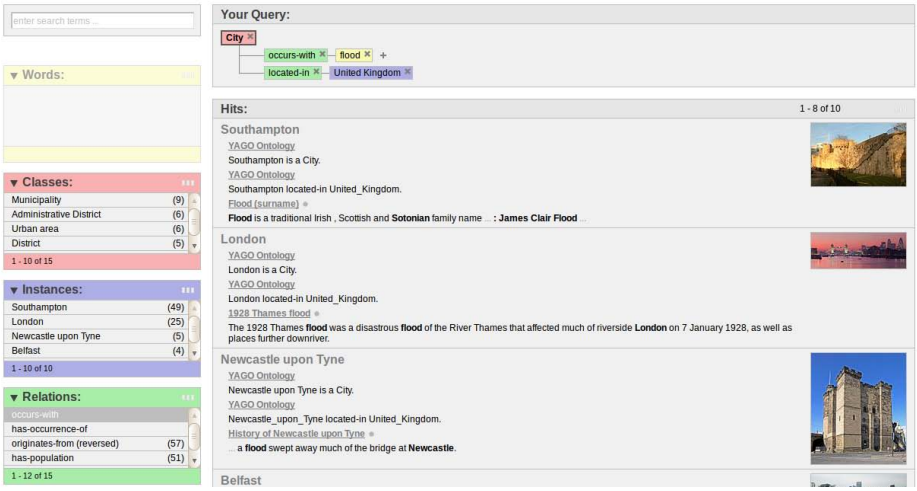

Fig. 5. The Broccoli Interactive Query Building UI

Figure 6 shows a query where the user has typed the word "ontology", which in GATE can refer to various concepts in the GATE domain ontology of Java classes and their documentation. Due to this ambiguity, the results are first a list of candidate concepts. Once the user selects the intended GATE concept, a list of matching documents is displayed, listed by type (Java code, forum post, etc). Their titles are clickable, which then shows the document content with the query terms highlighted (see the example Java code in the bottom right corner of Figure 6).

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 deduct 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 [49] 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 Figure 7 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.

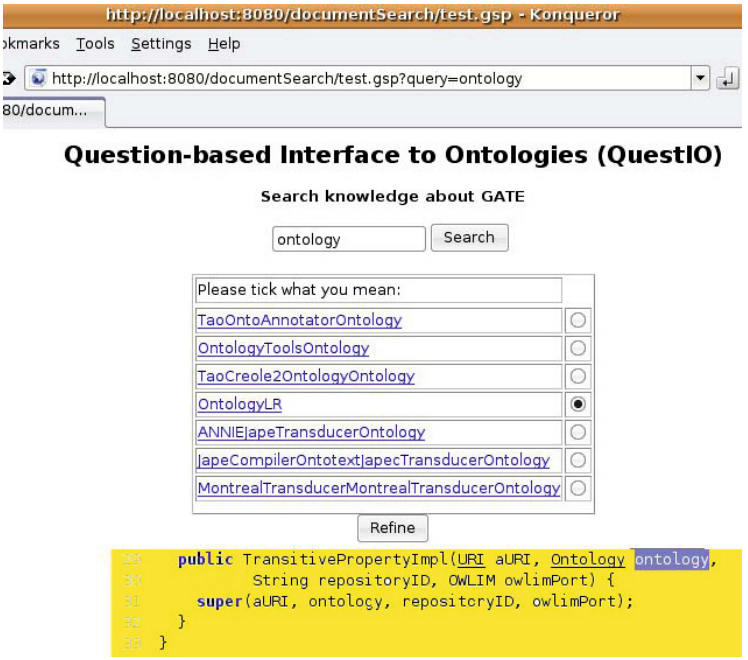

Fig. 6. A semantic search example in QuestIO

6.3 Form-Based Semantic Search Interfaces

One of the challenges faced by semantic search interfaces, especially in subject-specific cases, is to indicate to users what they can search for. A form-based interface makes this explicit, in a manner similar to the facet-based UIs discussed above.

One example of such interface is the EnviLOD UI (see Figure 8), which was developed as a user-friendly semantic search front end to a Mímir index of environmental science documents, terms, and LOD entities (DBpedia and GeoNames were used for this purpose).

There is a keyword search field, complemented with optional semantic search constraints, through a set of inter-dependent drop-down lists. In the first list, users can search for specific entity types (Locations, Organisations, Persons, Rivers, Dates) and Document – for specifying constraints on document-level attributes, etc. More than one semantic constraint can be added, through the plus button, which inserts a new row underneath the current row of constraints.

For instance, if Location is chosen as a semantic constraint, then, further constraints can be specified by choosing an appropriate property constraint (see Figure 8). Population allows users to pose restrictions on the population number of the locations that are being searched for. Similar numeric constraints can be imposed on the latitude, longitude, and population density attribute values.

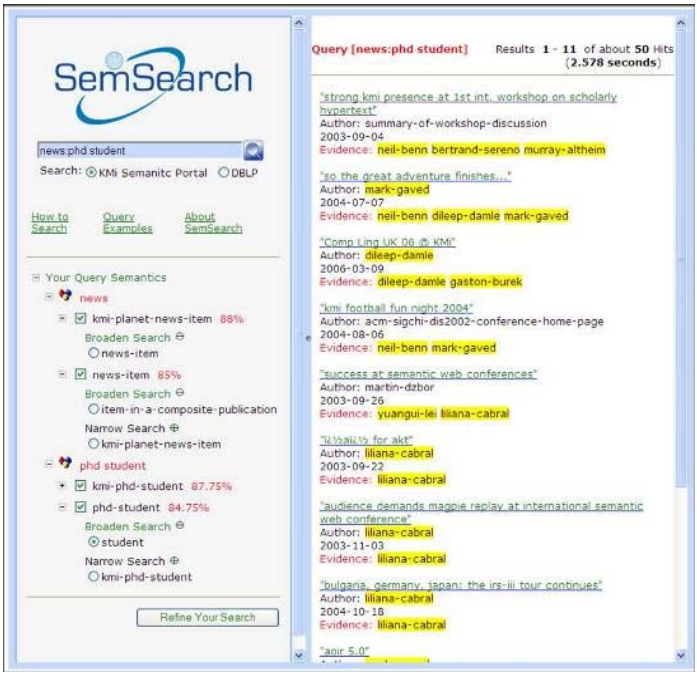

Fig. 7. Example SemSearch query results

Restrictions can also be imposed in terms of location name or the country it belongs to. For string-value properties, if "is" is chosen from the third list instead of none, then the value must be exactly as specified (e.g. Oxford), whereas "contains" triggers sub-string matching, (e.g. Oxfordshire is matched as a location name containing Oxford). In this way, a user searching for documents mentioning locations with name containing Oxford, will be shown not only documents mentioning Oxford explicitly, but also documents mentioning Oxfordshire and other locations in Oxfordshire (e.g. Wytham Woods, Banbury). In the latter case, the knowledge from DBpedia and GeoNames will be used to identify which other locations are in Oxfordshire, in addition to Oxford.

One problem with an EnviLOD-style UI is that it hides from users information about what instances of these classes occur in the indexed document collection, (e.g. which UK counties are mentioned). In order to provide such high-level entity-based overviews of the documents, one approach is to list all instances, for each class, as done in the KIM and Broccoli interfaces.

An alternative is to use tag clouds and other visualisations of entity co-occurrences. Mímir has recently been extended with such a user interface, called GATE Prospector (see Figure 9). The top half of the UI shows ontology classes and instances (UMLS in this example) and the user selects the desired ones. Additional search restrictions could be imposed via document metadata filters. The bottom half of Figure 9 shows the matching instances (terms in the case of

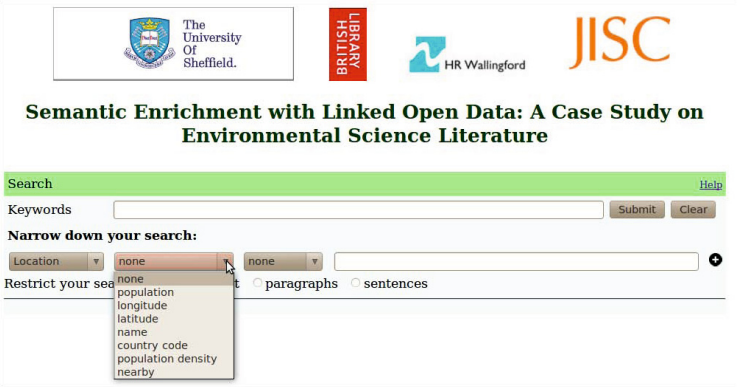

Fig. 8. The EnviLOD semantic search UI

UMLS) as well as the number of times they occur in the document collection. A frequency-based term cloud is also shown. The set of terms/instances can be saved for later use, e.g. to generate entity/term co-occurrence visualisations.

Figure 10 shows an example co-occurrence visualisation, where the most frequently mentioned instances of diseases are plotted against the most frequently mentioned instances of pathogens. Examples from other domains include plotting which sentiment terms co-occur most frequently with which political parties or politicians, given a large collection of tweets about an election.

7 Evaluation

In the context of evaluating semantic search over RDF triples, a repeatable and generic evaluation framework has emerged recently [50]. However, due to its focus purely on structured data, it is not directly reusable for evaluation of semantic full-text search.

Another example is [51], who again evaluate search over RDF facts, compared against web search. In this case, 200 queries are used and 12,000 relevance judgements are collected from human judges. Relevance feedback from a web-based keyword search is used to improve the results, however, the two types of search are kept entirely separate, instead of combined as in semantic full-text search. Therefore, again this dataset cannot be reused directly.

Specifically with respect to evaluating semantic full-text search, [34] use a TREC collection, 20 semantic search queries adapted from two TREC evaluations and the corresponding judgements, as well as a number of relevant ontologies and knowledge bases. Semantic search is compared against Lucene and TREC-based systems, evaluated using mean average precision. Query time performance is also reported.

The CE^2 system is evaluated using Wikipedia for the documents and DBpedia as the semantic knowledge base, thus directly exploiting the connection

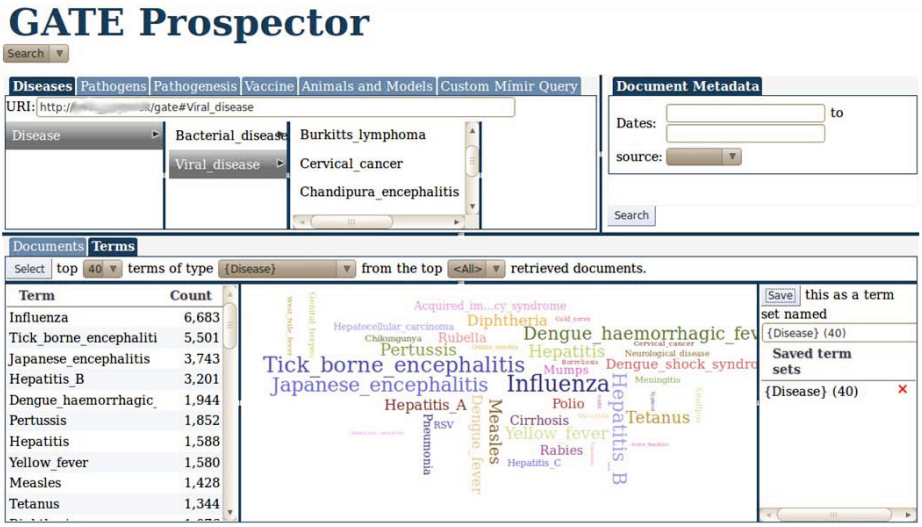

Fig. 9. The GATE Prospector semantic search UI

between the two [35] (links in Wikipedia articles are easily mapped to DBpedia URIs). The collection contains 2.1 million documents and 42 million annotations. The textual data and annotations are stored in Lucene, whereas the DBpedia triples (4.3 million) are stored in a relational database. 5 kinds of semantic full-text search queries are evaluated: retrieving documents containing an annotation with a specified instance URI; retrieving documents mentioning classes of annotations (e.g. companies); conjunctive queries requiring some reasoning with the RDF triples (e.g. institutions of scientists mentioned in a given document); user-created questions of the previous 3 types; questions based on Wikipedia list pages. Relevance judgements were produced for the fourth kind of queries by 20 researchers.

The most comprehensive and publicly available dataset (including queries and relevance judgements) is the one created for the evaluation of the Broccoli semantic full-text search system [21].[7] Similar to the previous system, it uses Wikipedia, but combined with the YAGO ontology, instead of DBpedia. A total of 61 queries were used (including 15 from a TREC benchmark), as well as 10 Wikipedia lists (similar to those used in the evaluation of CE^2). Traditional IR evaluation metrics are used, but no comparison against an IR baseline or another semantic search system is made on their dataset.

To summarise, evaluation of semantic full-text search is still in nascent stages, due to the lack of a shared evaluation task on which different approaches can be evaluated independently, in a reliable manner. Keyword-based IR systems are sometimes used as baselines, but, at the time of writing, no in-depth comparisons between semantic full-text search systems have been published.

[7] `http://broccoli.informatik.uni-freiburg.de/repro-corr/`

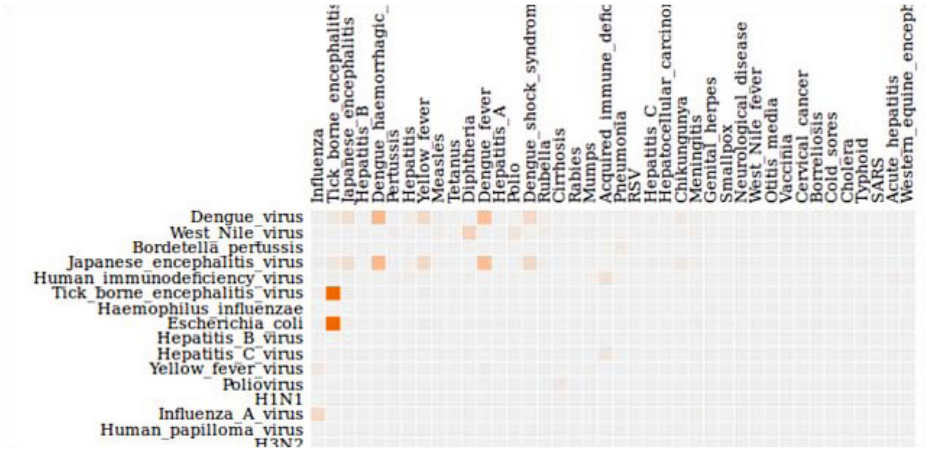

Fig. 10. GATE Prospector: Instance/Term Co-occurrence View

8 Conclusions and Future Work

This chapter provided a high-level introduction to semantic search over documents and ontologies. Semantics is needed in particular in cases where the correct interpretation of the search query requires knowledge not contained explicitly in the full text documents. For example, documents about flooding in UK cities with population under 50,000 people.

In summary, there seems to be an emerging consensus that current keyword-based, full-text search does not cater sufficiently well for information discovery. Semantic search and Linked open data offer much promise, but require significant development effort to implement in a way, which makes them accessible to end-users. Present prototypes typically require some understand of the underlying ontological relations, which give rise to the semantic search properties, which is a limitation to wider take-up.

Evaluations of semantic search (e.g. [52]) have demonstrated the need to show more feedback on why a certain document was matched, especially when this is a result of using implicit semantic knowledge from external knowledge resource. For example, in the EnviLOD interface, if a population or a distance-based constraints are used, relevance was not always obvious to the user.

One of the challenges with adopting semantic search is making the user interfaces as intuitive as possible, in order to reduce the initial learning curve. Particular attention needs to be paid to usability and design details, especially consistency with keyword-based general-purpose search engines, such as Google, Yahoo. These details should not be underestimated, even in research prototypes, since they can influence heavily the evaluation results.

Secondly, scalability and computational efficiency of semantic search are non-trivial, especially combining large volumes of content and entities from large LOD datasets.

The trade-off between precision and recall of the methods also could benefit from in-depth investigation and tuning, as well as showing explicitly an indicator of system confidence in the accuracy of the results. Existing semantic search frameworks could also benefit from taking into account user feedback on the relevance of each returned result, to help train the retrieval algorithms underneath.

Fourthly, current semantic query languages do not support negation in queries, which is a search feature required in many circumstances. Similarly, user adoption could be improved through adding a "more like this" functionality, in order to help users with refining semantic search queries.

Acknowledgements. This work was supported by funding from the Engineering and Physical Sciences Research Council (grant EP/I004327/1), JISC Research Tools (#restools) Strand B3 project 2658 (EnviLOD), and the Trendminer project, EU FP7-ICT Programme, grant agreement no.287863.

References

1. Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., Kirilov, A., Goranov, M.: Semantic annotation, indexing and retrieval. Journal of Web Semantics, ISWC 2003 Special Issue 1(2), 671–680 (2004)
2. Cunningham, H., Tablan, V., Roberts, I., Greenwood, M.A., Aswani, N.: Information Extraction and Semantic Annotation for Multi-Paradigm Information Management. In: Lupu, M., Mayer, K., Tait, J., Trippe, A.J. (eds.) Current Challenges in Patent Information Retrieval. The Information Retrieval Series, vol. 29, pp. 307–327. Springer, Heidelberg (2011)
3. Mahesh, K., Kud, J., Dixon, P.: Oracle at TREC8: A Lexical Approach. In: Proceedings of the Eighth Text Retrieval Conference, TREC-8 (1999)
4. Voorhees, E.: Using WordNet for Text Retrieval. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database. MIT Press (1998)
5. Gruber, T.R.: A Translation Approach to Portable Ontologies. Knowledge Acquisition 5(2), 199–220 (1993)
6. Singhal, A.: Introducing the knowledge graph: things, not strings (May 2012)
7. Pound, J., Mika, P., Zaragoza, H.: Ad-hoc object retrieval in the web of data. In: Proceedings of the 19th International Conference on World Wide Web, pp. 771–780. ACM (2010)
8. Ratinov, L., Roth, D.: Design challenges and misconceptions in named entity recognition. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning, pp. 147–155. Association for Computational Linguistics (2009)
9. Bontcheva, K., Cunningham, H.: Semantic annotations and retrieval: Manual, semiautomatic, and automatic generation. In: Domingue, J., Fensel, D., Hendler, J. (eds.) Handbook of Semantic Web Technologies, pp. 77–116. Springer, Heidelberg (2011)
10. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: Proc. of the 17th Conf. on Information and Knowledge Management (CIKM), pp. 509–518 (2008)
11. Rao, D., McNamee, P., Dredze, M.: Entity linking: Finding extracted entities in a knowledge base. In: Multi-source, Multi-lingual Information Extraction and Summarization. Springer (2013)

12. Ji, H., Grishman, R.: Knowledge base population: Successful approaches and challenges. In: Proc. of ACL 2011, pp. 1148–1158 (2011)
13. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia Spotlight: Shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems, pp. 1–8 (2011)
14. Shen, W., Wang, J., Luo, P., Wang, M.: LINDEN: Linking named entities with knowledge base via semantic knowledge. In: Proceedings of the 21st Conference on World Wide Web, pp. 449–458 (2012)
15. Gruhl, D., Nagarajan, M., Pieper, J., Robson, C., Sheth, A.: Context and Domain Knowledge Enhanced Entity Spotting in Informal Text. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 260–276. Springer, Heidelberg (2009)
16. Kiryakov, A., Ognyanoff, D., Velkov, R., Tashev, Z., Peikov, I.: Ldsr: Materialized reason-able view to the web of linked data. In: OWL: Experiences and Directions workshop (OWLED 2009) (2009)
17. Klyne, G., Carroll, J.: Resource description framework (RDF): Concepts and abstract syntax. W3C recommendation, W3C (2004), http://www.w3.org/TR/rdf-concepts/
18. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL web ontology language reference. W3C recommendation, W3C (February 2004), http://www.w3.org/
19. Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF. W3C recommendation — 15 January 2008, W3C (2008), http://www.w3.org/, http://www.w3.org/TR/rdf-sparql-query/.
20. Bast, H., Bäurle, F., Buchhold, B., Haussmann, E.: A case for semantic full-text search. In: Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search, JIWES 2012, pp. 4:1–4:3. ACM (2012)
21. Bast, H., Bäurle, F., Buchhold, B., Haussmann, E.: Broccoli: Semantic full-text search at your fingertips. CoRR abs/1207.2615 (2012)
22. Kieniewicz, J., Sudlow, A., Newbold, E.: Coordinating improved environmental information access and discovery: Innovations in sharing environmental observations and information. In: Pillman, W., Schade, S., Smits, P. (eds.) Proceedings of the 25th International EnviroInfo Conference (2011)
23. Kieniewicz, J., Wallis, M.: User requirements. Technical Report, EnviLOD project deliverable (2012), http://gate.ac.uk/projects/envilod/EnviLOD-WP2-User-Requirements.pdf
24. Lupu, M., Hanbury, A.: Patent retrieval. Foundations and Trends in Information Retrieval 7(1), 1–97 (2013)
25. Haas, K., Mika, P., Tarjan, P., Blanco, R.: Enhanced results for web search. In: Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), pp. 725–734 (2011)
26. Cunningham, H., Tablan, V., Roberts, A., Bontcheva, K.: Getting more out of biomedical documents with gate's full lifecycle open source text analytics. PLoS Computational Biology 9(2), e1002854 (2013)
27. Li, Y., Bontcheva, K., Cunningham, H.: Hierarchical, Perceptron-like Learning for Ontology Based Information Extraction. In: 16th International World Wide Web Conference (WWW 2007), pp. 777–786 (May 2007)
28. McDowell, L.K., Cafarella, M.: Ontology-Driven Information Extraction with OntoSyphon. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 428–444. Springer, Heidelberg (2006)

29. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: Gate: an architecture for development of robust hlt applications. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL 2002, July 7-12, pp. 168–175. Association for Computational Linguistics, Stroudsburg (2002)
30. Bontcheva, K., Cunningham, H.: Semantic annotation and retrieval: Manual, semi-automatic and automatic generation. In: Domingue, J., Fensel, D., Hendler, J.A. (eds.) Handbook of Semantic Web Technologies. Springer (2011)
31. 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 Thirteenth ACM Conference on Information and Knowledge Management (2004)
32. Hildebrand, M., van Ossenbruggen, J., Hardman, L.: /facet: A Browser for Heterogeneous Semantic Web Repositories. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 272–285. Springer, Heidelberg (2006)
33. Zhang, L., Liu, Q., Zhang, J., Wang, H., Pan, Y., Yu, Y.: Semplore: An IR approach to scalable hybrid query of semantic web data. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 652–665. Springer, Heidelberg (2007)
34. Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., Motta, E.: Semantically enhanced information retrieval: An ontology-based approach. Web Semantics 9(4), 434–452 (2011)
35. Wang, H., Tran, T., Liu, C., Fu, L.: Lightweight integration of ir & db for scalable hybrid search with integrated ranking support. Web Semantics: Science, Services and Agents on the World Wide Web 9(4) (2011)
36. Fazzinga, B., Gianforme, G., Gottlob, G., Lukasiewicz, T.: Semantic web search based on ontological conjunctive queries. Web Semantics: Science, Services and Agents on the World Wide Web 9(4) (2011)
37. Bikakis, N., Giannopoulos, G., Dalamagas, T., Sellis, T.: Integrating keywords and semantics on document annotation and search. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6427, pp. 921–938. Springer, Heidelberg (2010)
38. Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., Goranov, M.: KIM – Semantic Annotation Platform. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 834–849. Springer, Heidelberg (2003)
39. Kiryakov, A.: OWLIM: balancing between scalable repository and light-weight reasoner. In: Proceedings of the 15th International World Wide Web Conference (WWW 2006), Edinburgh, Scotland, May 23-26 (2006)
40. Boldi, P., Vigna, S.: MG4J at TREC 2005. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005), November 15-18. Special Publications, NIST, vol. 500, pp. 266–271 (2005), `http://mg4j.dsi.unimi.it/`
41. Maynard, D., Greenwood, M.A.: Large Scale Semantic Annotation, Indexing and Search at The National Archives. In: Proceedings of LREC 2012, Turkey (2012)
42. Tablan, V., Roberts, I., Cunningham, H., Bontcheva, K.: Gatecloud.net: a platform for large-scale, open-source text processing on the cloud. Philosophical Transactions of the Royal Society A 371(1983) (2013)
43. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-Based Lookup through the User Interaction. In: Aroyo, L., Antoniou, G., Hyvönen, E., ten Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010, Part I. LNCS, vol. 6088, pp. 106–120. Springer, Heidelberg (2010)

44. Lopez, V., Uren, V., Motta, E., Pasin, M.: AquaLog: An Ontology-driven Question Answering System for Organizational Semantic Intranets. Web Semantics: Science, Services and Agents on the World Wide Web 5(2), 72–105 (2007)
45. Kaufmann, E., Bernstein, A.: How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users? In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 281–294. Springer, Heidelberg (2007)
46. Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., Davis, B., Handschuh, S.: CLOnE: Controlled Language for Ontology Editing. In: Aberer, K., et al. (eds.) ISWC/ASWC 2007. LNCS, vol. 4825, pp. 142–155. Springer, Heidelberg (2007)
47. Bernstein, A., Kaufmann, E.: GINO - A Guided Input Natural Language Ontology Editor. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 144–157. Springer, Heidelberg (2006)
48. Damljanovic, D., Bontcheva, K.: Enhanced Semantic Access to Software Artefacts. In: Workshop on Semantic Web Enabled Software Engineering (SWESE), Karlsruhe, Germany (October 2008)
49. Lei, Y., Uren, V.S., Motta, E.: SemSearch: A Search Engine for the Semantic Web. In: Staab, S., Svátek, V. (eds.) EKAW 2006. LNCS (LNAI), vol. 4248, pp. 238–245. Springer, Heidelberg (2006)
50. Blanco, R., Halpin, H., Herzig, D.M., Mika, P., Pound, J., Thompson, H.S., Tran, T.: Repeatable and reliable semantic search evaluation. Web Semantics: Science, Services and Agents on the World Wide Web (in press)
51. Halpin, H., Lavrenko, V.: Relevance feedback between hypertext and semantic web search: Frameworks and evaluation. Web Semantics: Science, Services and Agents on the World Wide Web 9(4) (2011)
52. Bontcheva, K., Kieniewicz, J., Aswani, N., Wallis, M., Andrews, S.: User feedback report on the envilod semantic search interface. Technical Report, EnviLOD project deliverable (2012), `http://gate.ac.uk/projects/envilod/EnviLOD-user-feedback-report.pdf`