Semantic Annotation, Indexing, and Retrieval

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

Ontotext Lab, Sirma AI EOOD, 138 Tsarigradsko Shose, Sofia 1784, Bulgaria
{naso,borislav,damyan,mitac,angel,miro}@sirma.bg

Abstract. The Semantic Web realization depends on the availability of critical mass of metadata for the web content, linked to formal knowledge about the world. This paper presents our vision about a holistic system allowing annotation, indexing, and retrieval of documents with respect to real-world entities. A system (called KIM), partially implementing this concept is shortly presented and used for evaluation and demonstration.

Our understanding is that a system for semantic annotation should be based upon specific knowledge about the world, rather than indifferent to any ontological commitments and general knowledge. To assure efficiency and reusability of the metadata we introduce a simplistic upper-level ontology which starts with some basic philosophic distinctions and goes down to the most popular entity types (people, companies, cities, etc.), thus providing many of the inter-domain common sense concepts and allowing easy domain-specific extensions. Based on the ontology, an extensive knowledge base of entities descriptions is maintained.

Semantically enhanced information extraction system providing automatic annotation with references to classes in the ontology and instances in the knowledge base is presented. Based on these annotations, we perform IR-like indexing and retrieval, further extended using the ontology and knowledge about the specific entities.

1 Introduction

Semantic Web is about adding formal semantics (metadata, knowledge) to the web content for the purpose of more efficient access and management. Since its vitality depends on the presence of critical mass of metadata, the acquisition of this metadata is a major challenge for the Semantic Web community. Though, in some cases unavoidable, the manual accumulation of this explicit semantics is not considered a feasible approach. Our vision is that fully automatic methods for semantic annotation should be researched and developed. For this to happen, the necessary design and modeling questions should be faced and resolved, and the enabling complementary resources and infrastructure should be provided. To assure wide acceptance and usage of semantic annotation systems their tasks should be clearly defined, their performance – properly evaluated and communicated.

The semantic annotation offered here is a specific metadata generation and usage schema targeted to enable new information access methods and extend existing ones. The annotation scheme offered is based on the understanding that the named entities

D. Fensel et al. (Eds.): ISWC 2003, LNCS 2870, pp. 484–499, 2003.

(NE, see 1.1) mentioned in the documents constitute important part of their semantics. Further, using different sorts of redundancy, external or background knowledge, those entities can be coupled with formal descriptions and thus provide more semantics and connectivity to the web. We hope that the expectations towards the Semantic Web will be easier to realize if the following basic tasks can be defined and solved:

1. Annotate and hyperlink (references to) named entities in text documents;

2. Index and retrieve documents with respect to the referred entities.

The first task can be seen as an advanced combination of basic press-clipping exercise, typical IE[1] task, and automatic hyper-linking. The resulting annotations represent basically a document enrichment and presentation method, which can further be used to enable other access methods.

The second task is just a modification of the classical IR task – documents are retrieved based on relevance to NEs instead of words. However the basic assumption is quite similar – the documents are characterized by the bag of tokens[2] constituting their content, disregarding its structure. While the basic IR approach considers as tokens the word stems, for the last decade there was considerable effort towards using word-senses or lexical concepts (see [20] and [36]) for indexing and retrieval. The named entities can be seen as special sort of token to be taken care of. What we present here is one more (pretty much independent) development direction instead of alternative of the contemporary IR trends.

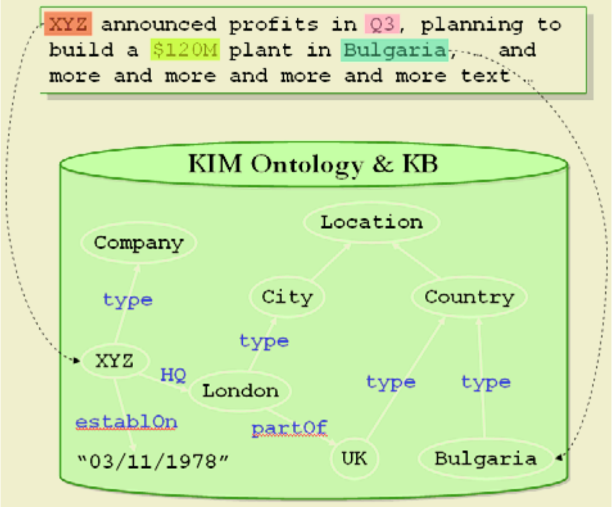

XYZ announced profits in Q3, planning to build a $120M plant in Bulgaria, and more and more and more and more text
KIM Ontology & KB
Location
Company
type
City
Country
type
XYZ
HQ
London
type
type
establOn
partOf
"03/11/1978"
UK
Bulgaria

Fig. 1. Semantic Annotation

In a nutshell, Semantic Annotation is about assigning to the entities in the text links to their semantic descriptions (as presented on Fig. 1). This sort of metadata provides

[1] Information extraction, a relatively young discipline in the Natural Language Processing (NLP), which conducts partial analysis of text in order to extract specific information, [6].

[2] Or "atomic text entities" as those are referred in [17].

both class and instance information about the entities. It is a matter of terminology whether these annotations should be called "semantic", "entity" or some other way. To the best of our knowledge there is no well-established term for this task; neither there is a well-established meaning for "semantic annotation". What is more important, the automatic semantic annotations enable many new applications: highlighting, indexing and retrieval, categorization, generation of more advanced metadata, smooth traversal between unstructured text and available relevant knowledge. Semantic annotation is applicable for any sort of text – web pages, regular (non-web) documents, text fields in databases, etc. Further, knowledge acquisition can be performed based on extraction of more complex dependencies – analysis of relationships between entities, event and situation descriptions, etc.

This paper presents a schema for automatic semantic annotation, indexing and retrieval, together with a discussion on number of design and modeling questions (section 2) followed by discussion on the process (section 3). In section 4, we present a software platform, KIM, which demonstrates this model based on the latest Semantic Web and Information Extraction technology. The fifth section provides survey on related work. Conclusion and future work are discussed in section 6.

1.1 Named Entities

In the NLP and particularly IE tradition, **named entities** are considered: *people, organizations, locations*, and others referred by name. In a wider interpretation, those include also scalar values (*numbers,dates, amounts of money)*, *addresses*, etc.

The NEs require different handling because of their different nature and semantics[3] as opposed to the words (terms, phrases, etc.) While the former denote particulars (individuals or instances), the later denote universals (concepts, classes, relations, attributes). While the words can be described with the means of lexical semantics and common sense, the understanding and managing of named entities, requires more specific world knowledge.

2 Semantic Annotation Model and Representation

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

- Ontology (or at least taxonomy) defining the entity classes. It should be possible to refer to those classes;
- Entity identifiers which allow those to be distinguished and linked to their semantic descriptions;
- Knowledge base with entity descriptions.

[3] Without trying to discuss what semantic means in general, we simplify it down to "a model or description of an object which allows further interpretation."

The next question considers an important choice for the representation of the annotations – "to embed or not to embed?" Although the embedded annotations seem easier to maintain, there are number of arguments providing evidence that the semantic annotations have to be decoupled from the content they refer to. One key reason is to allow dynamic, user-specific, semantic annotations – the embedded annotations become part of the content and may not change corresponding to the interest of the user or the context of usage. Further, embedded complex annotations would have negative impact on the volume of the content and can complicate its maintenance – imagine that page with three layers of overlapping semantic annotations need to be updated preserving them consistent. Those and number of other issues defending the externally encoded annotation can be found in [34] which also provides an interesting parallel to the open hypermedia systems.

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

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

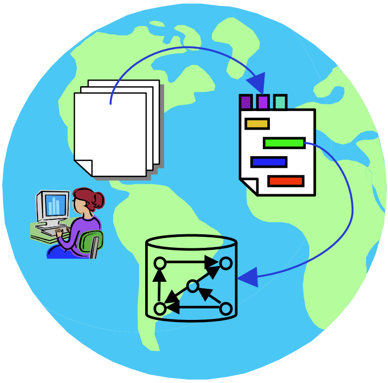


Fig. 2. Distributed Heterogeneous Knowledge

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

2.1 Light-Weight Upper Level Ontology

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

The ontology to support semantic annotation in a web context should address number of general classes which use to appear in texts in various domains. Describing these classes together with the most basic relations and attributes means that an upper-level ontology should be involved. The experience within number of projects[4] demonstrates that "logically extensive" upper-level ontologies are extremely hard to agree on, build, maintain, understand, and use. This seems to provide enough evidence that a light-weight upper level ontology is necessary for semantic annotations.

2.2 Knowledge Representation Language

According to the analysis of ontology and knowledge representation languages and formats in [11] and other authors it becomes evident that there is no much consensus beyond RDF(S), see [4]. The latter is well established in the Semantic Web community as a knowledge representation and interchange language. The rich diversity of RDF(S) repositories, APIs and tools, forms a mature environment for development of systems grounded in RDF(S) representation of their ontological and knowledge resources. Because of the common acceptance of RDF(S) in the Semantic Web community, it would be easy to reuse the ontology and KB, as well as enrich them with domain-specific extensions. The new OWL (see [9]) standard offers clear, relatively consensual and backward-compatible path beyond RDF(S), but still lacks sufficient tool support. Our experience shows (see the section on KIM) that for the basic purposes of light-weight ontology definition and entity description, RDF(S) provides sufficient basic expressiveness. The most critical nice-to-have primitives (equality, transitive and symmetric relations, etc.) are well covered in OWL Lite – the simplest first level of OWL. So, we suggest that RDF(S) is used in a way which allows easy extension towards OWL – this means avoiding primitives and patterns not included in OWL, http://www.w3.org/2002/07/owl.

2.3 Metadata Encoding and Management

The metadata has to be stored in a format allowing its efficient management; we are not going to prescribe a specific format here, but rather to outline number of principles and requirements towards the document and annotation management:

[4] For instance, Cyc (http://www.cyc.com) and the Standard Upper Ontology initiative (http://suo.ieee.org/)

- Documents (and other content) in different formats to be identifiable and their text content to be accessible;
- To allow non-embedded annotations over documents to be stored, managed and retrieved according to their positions, features, and references to a KB;
- To allow embedding of the annotations at least for some of the formats;
- To allow export and exchange of the annotations in different formats.

There are number of standards and initiatives related to encoding and representation of metadata related to text. Two of the most popular are TEI[5] and Tipster[6].

2.4 Knowledge Base

Once having the entity types, relations, and attributes encoded in an ontology, the next aspect of the semantic annotation representation are the entity descriptions. It should be possible to identify, describe and interconnect the entities in a general, flexible and standard fashion. We call a body of formal knowledge about entities a knowledge base (KB) – although a bit old-fashioned, this term reflects best the representation of non-ontological formal knowledge. A KB is expected to contain mostly instance knowledge/data, so, other names can also make a good fit for such dataset.

We consider that the ontology (defining all classes, relations and attributes, together with further constraints and dependencies) is a sort of schema for the KB and both should be kept into a semantic store – any sort of formal knowledge reasoning and management system which provides the basic operations: storage and retrieval according to the syntax and semantics of the selected formalism. The store may or may not provide inference[7], it can implement different reasoning strategies, etc. There are also more advanced management features which are not considered as a must: versioning, access control, transaction support, locking, client-caching. For an overview of those see [16], [15], [19] and [25]. Whether the ontology and knowledge base should be kept together – this is a matter of distributed knowledge representation and management which is outside the scope of this paper.

The KB can host two sorts of entity knowledge (descriptions and relationships):

- Pre-populated – such imported or otherwise acquired from trusted sources;
- Automatically extracted – such discovered in the process of semantic annotation (say via IE) or using other knowledge discovery and acquisition methods such as data-mining.

It is up to the specific implementation, whether or not and how much the KB to be pre-populated. For instance, information about entities of general importance (including their aliases) can significantly help the IE used for automatic semantic annotations – an extensive proposal about this can be found in the description of the KIM platform later on in this paper.

Further, domain and task specific knowledge could help the customization of a semantic annotation application – after extending the ontology to match the appliance

[5] The Text Encoding Initiative, http://www.tei-c.org/

[6] Tipster Architecture, http://www.cs.nyu.edu/cs/faculty/grishman/tipster.html

[7] For instance, there are experts who do not consider as inference the interpretation of RDF(S) according to its model-theoretic semantics, just because this one is simple compared to semantic and the inference methods in other languages.

domain, the KB could be pre-populated with specific entities. For instance, information about specific markets, customers, products, technologies and competitors could be of a great help for business intelligence and press-clipping; for company intelligence within UK it would be important to have more exhaustive coverage of UK-based companies and UK locations. It might also appear beneficial to reduce the general information that is not applicable in the concrete context and thus construct a more focused KB.

Since state of the art IE (and in particular named entity recognition, NER) allows recognition of new (previously unknown) entities and relations between them, it is reasonable to use this advantage for the enrichment of the KB. Because of the innate non-preciseness of these methods, the knowledge accumulated through them should be distinguishable from the one that was pre-populated. Thus the extraction of new metadata, can still be grounded in the trusted knowledge about the world, while the accumulated entities would be available for indexing, browsing and navigation. Recognized entities could be transformed to trusted ones at some point, through semi-automatic validation process. Important part of this enrichment would be the template extraction of entity relations, which could be referred to as some kind of content-based learning of the system. Depending on the texts that are being processed, the respective changes would occur in the recognized parts of the KB, and thus its projection of the world would change accordingly (e.g. processing only sport news articles, the metadata would be both rich for this domain and poor for the others.)

2.5 Unified Representation of Lexical Knowledge

The symbolic IE processing usually requires some lexica to be used for pattern recognition and other purposes. These are both general entries (such as various sorts of stop words) as well as such specific for the entity classes being handled. It is common that IE systems keep these in application-specific formats or directly hard-coded in the source code.

It's worth to represent and manage those in the same format used for the ontology and the entity knowledge base – this way the same tools (parsers, editors, etc.) can be used to manage both sorts of knowledge. For this purpose, part of the ontology (or just a separate one) could be dedicated to defining the types of lexical resources used by the natural language technologies involved.

The corresponding lexical resources part of the KB should be pre-populated to aid the IE process by providing clues for the entity and relation recognition, which goes beyond the already known instances. For instance, for efficient recognition of persons in the text one would need lists of first names (male and female), person titles, positions and professions. Some of these could be ontologically distinguishable by gender, as well. For the Organization lexica one should pre-populate possible suffixes (such as Ltd., GmbH, etc.), and terms appearing in the organization name (e.g. company, theatre, etc.). Additionally, time and date lexica ("a.m.", "Tue", etc.), currency units, address lexica and others should be included. The mature symbolic NER and IE systems already have coverage of such resources; the next step to integrate them in a system for automatic semantic annotation would be just to encode them in a formal ontology and present them in the KB.

3 Semantic Annotation Process

As already mentioned, we focus mainly on the automatic semantic annotation, leaving manual annotation to approaches more related to authoring web content. Even less accurate, the automatic approaches for metadata acquisition promise scalability and without them the Semantic Web will remain mostly a vision for long time. Our experience shows that the existing state-of-the-art IE systems have the potential to automate the annotation with reasonable accuracy and performance.

Although a lot of research and development contributed in the area of automatic IE so far, the lack of standards and integration with formal knowledge management systems was obscuring its usage. We claim that it is crucial to encode the extracted knowledge formally and according to well known and widely accepted knowledge representation and metadata encoding standards. Such system should be easily extensible for domain-specific applications, providing basic means for addressing the most common entities types, their attributes, and relations.

3.1 Extraction

It is a major problem with the traditional NER approaches that the annotations produced are not encoded in an open formal system and unbound entity types are used. The resources used are also traditionally presented in a proprietary form with no clear semantics. This hinders the reuse of both lexical resources and the resulting annotations by other systems, thus limiting the progress of the language technologies, since effortless sharing of resources and results is too expensive.

These problems can be partly resolved by an ontology-based infrastructure for IE. As proposed above, the entity types should be defined within an ontology, and the entities being recognized to be described (or at least kept) in accompanying KB. Thus the NLP systems with ontology support would more easily share both pre-populated knowledge and the results of their processing, as well, as all the different sorts of lexicons and other resources commonly used.

An important case demonstrating how ontologies can be used in IE are the so-called gazetteers used to look-up in the text predefined strings out of predefined lists. At present, the lists are being kept in proprietary formats. Typical result of the work of the gazetteers are annotations with some unbound strings used as types. A better approach presumes all the various annotation types and list values to be kept in a semantic store. Thus, the resulting annotation can be typed by reference to ontology classes and even further, point to specific lexeme or entity, if appropriate.

Since a huge amount of NLP research has been contributed in the recent years (and even decades), we suggest the reuse of existing systems with proven maturity, and effectiveness. Such system should be modified so to use resources kept in a KB and produce annotations referring to the latter. Our experience shows that such a change is not a trivial one. All the processing layers have to be re-engineered in order to get opened towards the semantic repository and depend on it for their inputs. However, there are number of benefits of such approach:

- All the various sorts of resources can be managed in a much more standard and uniform way;
- It becomes easier to manage the different sorts of linguistic knowledge at the proper level of generality. For instance, a properly structured entity type hierarchy

would allow that the entities and their references in the text are classified in the most precise way, but still easily matched in more general patterns. Thus, one can have a specific mountain annotated and still match it within a grammar rule which expects any sort of location;

- Wherever it is possible, any available further knowledge will be accessible directly with a reference from the annotation to the semantic store. Thus, available knowledge for an entity can be used for instance for disambiguation or co-reference resolution tasks.

A processing layer that is not inherent to the traditional IE systems can generate and store in the KB the descriptions of the newly discovered entities. When the same entity is encountered in the text next time, it can be directly linked to the already generated description. Further, extending the IE task to cover template relations extraction, another layer could enrich the KB with these relations.

3.2 Indexing and Retrieval

Historically, the issue of specific handling of the named entities was neglected by the information retrieval (IR) community, apart from some shallow handling for the purpose of Questions/Answering tasks. However, a recent large scale human interaction study on a personal content IR system of Microsoft (reported in [10]) demonstrates that, at least in some cases, the ignorance of the named entities does not match the user needs: "The most common query types in our logs were People/places/things, Computers/internet and Health/science. In the People/places thing category, names were especially prevalent. Their importance is highlighted by the fact that 25% of the queries involved people's names suggesting that people are a powerful memory cue for personal content. In contrast, general informational queries are less prevalent."

As the web content is rapidly growing, the demand of more advanced retrieval methods increases accordingly. Based on semantic annotations, efficient indexing and retrieval techniques could be developed involving explicit handling of the named entity references.

In a nutshell, the semantic annotations could be used to index both "NY" and "N.Y." as occurrence of the specific entity "New York" like if there was just its unique ID. Because of no entity recognition involved, the present systems will index on "NY", "N", and "Y" which demonstrates well some of the problems with the keyword-based search engines.

Given metadata indexing of the content, advanced semantic querying should be feasible. In a query towards a repository of semantically annotated documents, it should be possible to specify entity type restrictions, name and other attribute restrictions, as well as relations between the entities of interest. For instance, it should possible to make a query that targets all documents that refer to Persons that hold some Positions within an Organization, and also restricts the names of the entities or some of their attributes (e.g. a person's gender).

Further, semantic annotations could be used to match specific references in the text to more general queries. For instance, a query such as "company 'Redwood Shores'" could match documents mentioning the town and specific companies such as ORACLE and Symbian, but not the word "company".

Finally, although the above sketched enhancements look prominent, it still requires a lot of research and experiments to determine to what extent and how they could improve the existing IR systems. It is hard in a general context to predict how semantic indexing will combine with the symbolic and the statistical methods currently in use, such as the lexical approach presented in [20] and the latent semantic analysis presented in [18]. For this purpose, large scale experimental data and evaluation are required.

4 KIM Platform: Implementing the Vision

The Knowledge and Information Management (KIM) platform embodies our vision of semantic annotation, indexing and retrieval services and infrastructure. An essential idea in KIM, is the semantic (or entity) annotation, (as depicted on Fig. 1). It can be seen as a classical named-entity recognition and annotation process. However, in contrast to most of the existing IE systems, KIM provides for each entity reference in the text (i) a link (URI) to the most specific class in the ontology and (ii) a link to the specific instance in the knowledge base. The latest is (to the best of our knowledge) an unique KIM feature which allows further indexing and retrieval of documents with respect to entities.

For the end-user, the usage of a KIM-based application is straightforward and simple – requesting annotation from a browser plug-in, which highlights the entities in the current content and generates a hyperlink used for further exploring the available knowledge for the entity (as shown in Fig. 4). A semantic query web UI allows specification of a search query, that consists of entity type, name, attribute and relation restrictions (allowing queries such as **`Organization-locatedIn-Country`**, **`Person-hasPosition-Position-within-Organization,`** etc.)
This section provides a short overview of the main components of KIM, which is presented in bigger details in [27] and on its web site, http://www.ontotext.com/kim.

4.1 KIM Architecture

The KIM platform consists of KIM Ontology (KIMO) [8], knowledge base, KIM Server (with API for remote access, embedding, and integration), and front-ends (browser plug-in for Internet Explorer, Semantic Query web user interface, and Knowledge Explorer for KB navigation). KIM ontologies and knowledge bases are kept in the Sesame[9] RDF(S) repository and the Ontology Middleware Module[10] [16].

KIM provides a mature infrastructure for IE, annotation and document management, based on GATE[11] [7]. The Lucene[12] information retrieval engine has been adopted to index documents by entity types and measure relevance according

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

[9] http://sesame.aidministrator.nl/, RDF(S) repository by Aidministrator b.v.

[10] OMM (www.ontotext.com/omm) is an enterprise back-end for knowledge management.

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

[12] Lucene, http://jakarta.apache.org/lucene/, high performance full text search engine

entities, along with tokens and stems. It is important to mention that KIM, as a software platform, is domain and task independent as are Gate, Sesame and Lucene.

4.2 KIM Ontology (KIMO)

KIM uses a simplistic upper-level ontology starting with some basic philosophic distinctions between entity types (such as **`Object-`**s - existing entities such as locations and agents, **`Happening-`**s – defining events and situations, and **`Abstract-`**ions that are neither objects, neither happenings). Further on, the ontology goes in more details to such extent that real-world entity types of general importance are included (meetings, military conflicts, employment positions, commercial, government and other organizations, people, and various locations, etc.). The characteristic attributes and relations for the featured entity types, are defined (e.g. **`subRegionOf`** property for **`Location-`**s, **`hasPosition`** for **`Person`**s, **`locatedIn`** for organizations, etc.) Having this simplistic upper-level ontology as basis, one could add domain-specific extensions to it easily, for profiling the semantic annotation for concrete applications.

The distribution of the most commonly referred entity types varies greatly from domain to domain. As researched in [23], despite the difference of type distributions, there are several general entity types that appear in all corpuses – Person, Location, Organization, Money (amount), Dates, etc. The proper representation and positioning of those basic types was one of the objectives behind the design of KIMO. Further the ontology defines more specific entity types (e.g. **`Mountain`**, as a more specific type than **`Location`**.) The extent of specialization of the ontology is determined on the basis of research of the entity types in a corpus of general news (incl. political, sport, financial, etc.)

The KIM ontology (KIMO)[13] consists of about 250 classes and 100 properties. The top Entities could be seen in the type hierarchy of the KIM plug-in on Fig 4. The ontology is encoded in RDF(S). In addition number of "generative" (in the style of the RDFS MT semantics) axioms are defined, such as:

`<X, locatedIn, Y>` and **`<Y, subRegionOf, Z>`** => **`<X, locatedIn, Z>`**

This sort of axioms are supported by Sesame and provide easy to understand and manage consistent mechanism for "custom" extensions to the RDF(S) semantics with respect to specific ontology. Those axioms can be seen as an add-hoc but quite practical way to avoid the RDF(S) constraints without a need to implement some specific flavor of OWL or another language.

4.3 KIM Knowledge Base

The entity descriptions are being stored in the same RDF(S) repository as the KIM ontology. Each entity has information about its specific type, aliases (incl. a main alias, expressing the (most probable) official name), attributes (e.g. latitude of a **`Location`**), and relations (e.g. a **`Location subRegionOf`** another **`Location`**). A simplified schema of the entity representation is depicted on Fig. 3.

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

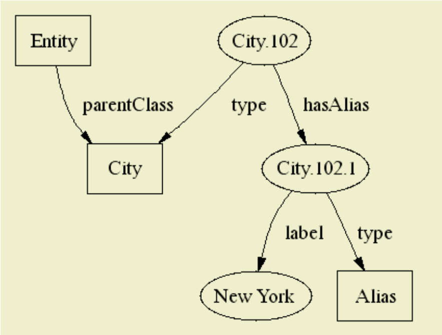


Fig. 3. Simplified entity description

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

4.4 KIM Information Extraction

KIM IE is based on the GATE framework, which has proved its maturity, extensibility and task independency for IE and other NL applications. The essence of the KIM IE, is the recognition of named entities (NE) with respect to KIMO ontology. The entity instances all bear unique identifiers (URI) that allow annotations to be linked both to the entity type and to the exact individual in the KB. For new (previously unknown) entities, URIs are being generated and assigned, next minimal descriptions are stored in the semantic store. The annotations are kept separated from the annotated content, and an API for their management is provided.

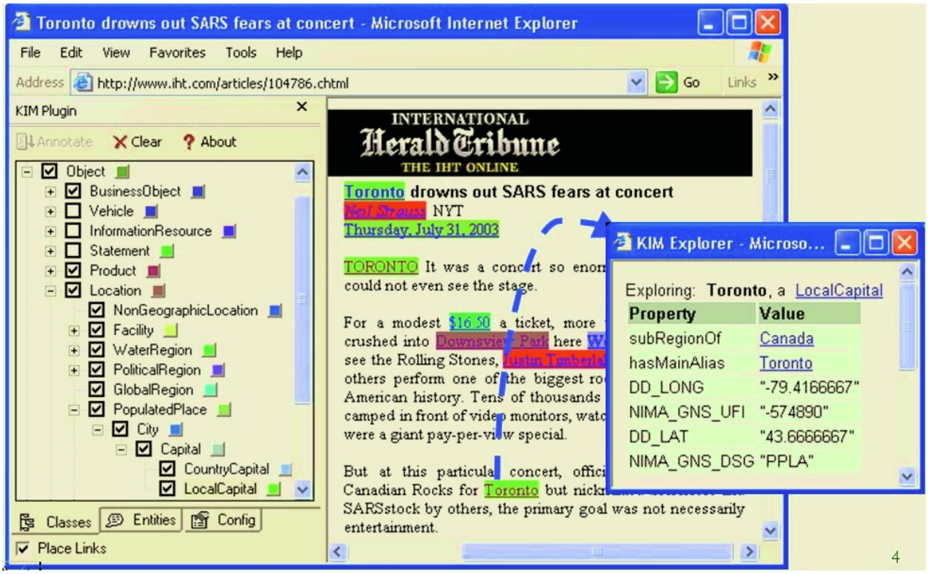
The actual processing of the content goes through several steps, starting with tokenization, splitting to sentences, and part of speech tagging. These processing layers are provided by the GATE framework, along with grammars and other standard building bricks for construction of sophisticated IE applications. However, number of components and resources have been considerably re-engineered and new ones were developed.

4.5 Indexing and Retrieval

Once the NER process has finished, the content is indexed with respect to specific NE. This enables queries with restrictions over entities, entity types, names, attributes, and relations. Technically, Lucene is adapted to perform full-text indexing, which is uniquely addressing each entity disregarding the alias used in the text. The retrieval accuracy of KIM has not been evaluated against a traditional IR engine, and this is a topic that would be researched in the future.

4.6 KIM Front-Ends

Different KIM front-end user interfaces are possible given the KIM API, which provides the functionality and infrastructure for the semantic annotation, indexing and retrieval, as well as document management, and KB navigation. We have created a



Toronto drowns out SARS fears at concert
Neil Strauss NYT
Thursday, July 31, 2003

TORONTO It was a conc rt so enor
could not even see the stage.

For a modest $16.50 a ticket, more
crushed into Downsview Park here W
see the Rolling Stones, Justin Timberla
others perform one of the biggest ro
American history. Tens of thousands
camped in front of video monitors, watc
were a giant pay-per-view special.

But at this particular concert, offici
Canadian Rocks for Toronto but nick
SARSstock by others, the primary goal was not necessarily
entertainment.

Exploring: Toronto, a LocalCapital

Property	Value
subRegionOf	Canada
hasMainAlias	Toronto
DD_LONG	"-79.4166667"
NIMA_GNS_UFI	"-574890"
DD_LAT	"43.6666667"
NIMA_GNS_DSG	"PPLA"

Fig. 4. The KIM plug-in with the top of the KIM Ontology, and KIM Explorer on top

plug-in (Fig. 4) for the Internet Explorer browser. The KIM plug-in provides lightweight semantic annotations delivery to the end user. On its first tab, the plug-in displays the entity type hierarchy (a branch of the KIM ontology). For each entity type there is an associated color used for highlighting the annotations of this type. Check boxes for each entity, allow the user to select the entity types of interest.

Upon invoking annotation of the current browser content, the plug-in extracts the text of the currently displayed document and sends it to an Annotation Server which is in its turn using the KIM Server NER API. The servers return the annotations with their offsets, type and instance information. The annotations are highlighted in the content (in the color of the respective entity type), and are hyperlinked to the KIM KB Explorer (Fig. 4). On the second tab of the plug-in there is a list of all the recognized entities for the current document, sorted by appearance frequency. Upon choosing from the list of entities, or following a hyperlink over an annotated entity in the text the user invokes the KIM KB Explorer, which provides a view of the part of the KB and the ontology that are related to the chosen entity (incl. type, aliases, relations and attributes). This way the user can directly navigate from the annotations to the instances that they are linked to in the KB. Via this explorer, the KB could be further explored by choosing one of the related entities, or the entity class.

5 Related Work

Semantic annotation of documents with respect to ontology and entity knowledge base is discussed in [5] and [14] – although presenting interesting and ambitious approaches, these do not discuss usage of information extraction for automatic annotation. The focus of [14] is manual semantic annotation for authoring web content, while [5] targets the creation of a web-based open hypermedia linking service, backed by a conceptual model of document terminology. Semantic annotation is used also in the S-CREAM project presented in [13] – the approach there is interesting with the heavy involvement of machine learning techniques for extraction of relations between the entities being annotated. Similar approach is taken also within the MnM project [35], where the semantic annotations can be placed inline in the document content and refer to an ontology and KB server (WebOnto), accessible via standard API. Another related approach is taken in OFF, [8], which puts an emphasize on the collaborative ontology development and annotation.

An interesting NE indexing and question/answering system is presented in [24]. Flat set of entity types is assigned to tokens and the annotations are incorporated in the content, in order to index by NE type later. Once indexed the content is queried via NL questions, with NE tagging over the question used to determine the expected answer type (e.g. When have the UN been established; UN here would be tagged with _ORG, specifying that the expected answer type is organization.) All the semantic annotation techniques above lack usage of upper-level ontologies and critical mass of world knowledge to serve as a trusted and reusable basis for the automatic recognition and annotation, as in the approach presented in [1] and discussed later on here.

Significant amount of research on information extraction (IE) has been performed in various projects within the GATE framework (see [6-7], [23]) with many existing tools and resources available. We build on those to provide language technology open to the Semantic Web standards and tools.

6 Conclusion and Future Work

This paper presented the notion of semantic annotation– an original meta-data model allowing ontology based named-entity annotation, indexing, and retrieval. Number of issues related to the representation and the usage of the semantic annotation were addressed. The KIM platform (addressed with more details in [27]) was shortly introduced to demonstrate an implementation of this vision.

The evaluation work done until now does not provide enough evidence regarding the approach, technology, and resources being used. The major obstacle is that there are neither test data nor well developed metrics for semantic annotation and retrieval.

Although naïve in some aspects, KIM platform provides a test bed and proves number of hypothesis and design decisions:

- It is worth using massive entity knowledge for. Even without comprehensive disambiguation, the precision drawbacks seem acceptable;
- It is possible to store and query tens of thousands of entities together with their descriptions in an RDF(S) repository (namely, Sesame);
- A simple but efficient technique for entity-aware IR is demonstrated;
- Few of light-weight front end tools can deliver in intuitive fashion the results of semantic annotation, indexing, and retrieval.

The challenges towards the general approach can be summarized as follows:

- Develop (or adapt) evaluation metric which properly measures the performance of a semantic annotation system;
- Experiment different approaches towards disambiguation of named-entity references: adaptation of a Hidden Markov Model learner successfully used for

non-semantic disambiguation is one of the first ideas; techniques similar to those used for word-sense disambiguation (namely, lexical-chaining); techniques for "symbolic" context management.

References

1. Bontcheva K., Kiryakov A., Cunningham H., Popov B., Dimitrov M. *Semantic Web Enabled, Open Source Language Technology.* In proc. of EACL Workshop "Language Technology and the Semantic Web", NLPXML-2003, 13 April, 2003
2. Brickley D, Guha R.V., eds. *Resource Description Framework (RDF) Schemas,* W3C http://www.w3.org/TR/2000/CR-rdf-schema-20000327/
3. Carr L., Bechhofer S., Goble C., Hall W. *Conceptual Linking: Ontology-based Open Hypermedia.* In The WWW10 Conference, Hong Kong, May, pp. 334–342.
4. Cunningham H., *Information Extraction: a User Guide* (revised version). Department of Computer Science, University of Sheffield, May, 1999
5. Cunningham H., Maynard D., Bontcheva K. and Tablan V., *GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications*. In proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics, 2002.
6. Collier N., Takeuchi K, Kawazoe A. *Open Ontology Forge: An Environment for Text Mining in a Semantic Web World.* In proc. of the International Workshop on Semantic Web Foundations and Application Technologies, Nara, Japan, 11th March.
7. Dean M., Connolly D., van Harmelen, F., Hendler J., Horrocks I., McGuinness D., Patel-Schneider P., Stein L.A., *Web Ontology Language (OWL) Reference Version 1.0.* W3C Working Draft 12 Nov. 2002, http://www.w3.org/TR/2002/WD-owl-ref-20021112/
8. Dumais S., Cutrell E., Cadiz J., Jancke G., Sarin R. and Robbins D. *Stuff I've Seen: A system for personal information retrieval and re-use.* In proc. of SIGIR'03, July 28 – August 1, 2003, Toronto, Canada, ACM Press, pp. 72–79.
9. Fensel D. *Ontology Language, v.2 (Welcome to OIL)* . Deliverable 2, On-To-Knowledge project, Dec 2001. http://www.ontoknowledge.org/downl/del2.pdf
10. Handschuh S., Staab St., Ciravegna F. *S-CREAM – Semi-automatic CREAtion of Metadata.* The 13th International Conference on Knowledge Engineering and Management (EKAW 2002), ed Gomez-Perez, A., Springer Verlag, 2002.
11. Kahan J., Koivunen M., Prud'Hommeaux E., Swick R. *Annotea: An Open RDF Infrastructure for Shared Web Annotations*. In The WWW10 Conference, Hong Kong, May, pp. 623–632.
12. Kampman A., Harmelen F., Broekstra J. *Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema*. In proc. of ISWC2002, June 9–12th, 2002, Italia.
13. Kiryakov A., Simov K. Iv., Ognyanov D. *Ontology Middleware: Analysis and Design* Del. 38, On-To-Knowledge, March 2002. http://www.ontoknowledge.org/downl/del38.pdf
14. Kiryakov A., Simov K. Iv. *Ontologically Supported Semantic Matching*. In proc. of "NODALIDA'99: Nordic Conference on Comp. Linguistics", Trondheim, Dec. 9–10, 1999.
15. Landauer T., and Dumais S. *A solution to Plato's problem: the Latent Semantic Analysis theory of acquisition, induction and representation of knowledge.* Psychological Review, 104(2), 1997, 211–240.
16. Maedche A., Motik B., Stojanovic L., Studer R. and Volz R. *Ontologies for Enterprise Knowledge Management* [http://kaon.semanticweb.org/docus/ieee-is-maedcheetal.pdf]. In IEEE Intelligent Systems, Vol. 18, Num. 2, pp. 26–33, 2003.
17. Mahesh K., Kud J., Dixon P. *Oracle at TREC8: A Lexical Approach*, In proc. of the Eighth Text Retrieval Conference (TREC-8), 1999.

18. Manov D, Kiryakov A, Popov B, Bontcheva K, Maynard D, Cunningham H. *Experiments with geographic knowledge for information extraction*. NAACL-HLT 2003, Canada. Workshop on the Analysis of Geographic References, May 31 2003, Edmonton, Alberta.
19. Maynard D., Tablan V., Bontcheva K., Cunningham H, and Wilks Y. *MUlti-Source Entity recognition – an Information Extraction System for Diverse Text Types*. Technical report CS–02–03, Univ. of Sheffield, Dep. of CS, 2003. http://gate.ac.uk/gate/doc/papers.html
20. Moldovan D., Mihalcea R. *Document Indexing Using Named Entities.* In "Studies in Informatics and Control", Vol. 10, No. 1, March 2001.
21. Noy N., Musen M. *Ontology Versioning as an Element of an Ontology-Management Framework*. IEEE Intelligent Systems, to appear, 2003.
22. Popov B., Kiryakov A., Kirilov A., Manov D., Ognyanoff D., Goranov M. *KIM – Semantic Annotation Platform.* In proc. of 2nd International Semantic Web Conference (ISWC2003), 20–23 October 2003, Florida, USA. To appear.
23. Pustejovsky J., Boguraev B., Verhagen, M., Buitelaar P., and Johnston M., *Semantic Indexing and Typed Hyperlinking*. In proc. of the AAAI Conference, Spring Symposium, NLP for WWW, Stanford University, CA, 1997, pp. 120–128.
24. van Ossenbruggen J., Hardman L., Rutledge L., *Hypermedia and the Semantic Web: A Research Agenda*. Journal of Digital information, volume 3 issue 1, May 2002.
25. Vargas-Vera M., Motta E., Domingue J., Lanzoni M., Stutt A. and Ciravegna F, *MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup*, In Proc. Of EKAW 2002, ed. Gomez-Perez, A., Springer Verlag, 2002.
26. 36. Voorhees E. *Using WordNet for Text Retrieval*. In "WordNet: an electronic lexical database." Fellbaum, C. (editor), MIT Press, 1998.