EU-IST Project IST-2003-506826 SEKT

SEKT: Semantically Enabled Knowledge Technologies

D2.6.1 Massive Automatic Annotation

Dimitar Manov, Borislav Popov

Ontotext Lab, Sirma Group Corp.

Abstract

This deliverable presents a software prototype, the first version of the Massive Automatic Annotation component, based on the KIM platform.

Keyword list: Massive Automatic Annotation, PROTON, KIM.

WP2 Human Language Technologies

Prototype	PU
Contractual date of delivery	M24

SEKT Consortium

This document is part of a research project partially funded by the IST Programme of the Commission of the European Communities as project number IST-2003-506826.

British Telecommunications plc.
Orion 5/12, Adastral Park
Ipswich IP5 3RE
UK
Tel: +44 1473 609583, Fax: +44 1473 609832
Contact person: John Davies
E-mail: john.nj.davies@bt.com

Empolis GmbH
Europaallee 10
67657 Kaiserslautern
Germany
Tel: +49 631 303 5540
Fax: +49 631 303 5507
Contact person: Ralph Traphöner
E-mail: ralph.traphoener@empolis.com

Jozef Stefan Institute
Jamova 39
1000 Ljubljana
Slovenia
Tel: +386 1 4773 778, Fax: +386 1 4251 038
Contact person: Marko Grobelnik
E-mail: marko.grobelnik@ijs.si

University of Karlsruhe, Institute AIFB
Englerstr. 28
D-76128 Karlsruhe
Germany
Tel: +49 721 608 6592
Fax: +49 721 608 6580
Contact person: York Sure
E-mail: sure@aifb.uni-karlsruhe.de

University of Sheffield
Department of Computer Science
Regent Court, 211 Portobello St.
Sheffield S1 4DP
UK
Tel: +44 114 222 1891
Fax: +44 114 222 1810
Contact person: Hamish Cunningham
E-mail: hamish@dcs.shef.ac.uk

University of Innsbruck
Institute of Computer Science
Techikerstraße 13
6020 Innsbruck
Austria
Tel: +43 512 507 6475
Fax: +43 512 507 9872
Contact person: Jos de Bruijn
E-mail: jos.de-bruijn@deri.ie

Intelligent Software Components S.A.
Pedro de Valdivia, 10
28006
Madrid
Spain
Tel: +34 913 349 797
Fax: +49 34 913 349 799
Contact person: Richard Benjamins
E-mail: rbenjamins@isoco.com

Kea-pro GmbH
Tal
6464 Springen
Switzerland
Tel: +41 41 879 00
Fax: 41 41 879 00 13
Contact person: Tom Bösser
E-mail: tb@keapro.net

Ontoprise GmbH
Amalienbadstr. 36
76227 Karlsruhe
Germany
Tel: +49 721 50980912
Fax: +49 721 50980911
Contact person: Hans-Peter Schnurr
E-mail: schnurr@ontoprise.de

Sirma Group Corp., Ontotext Lab
135 Tsarigradsko Shose
Sofia 1784
Bulgaria
Tel: +359 2 9768 303, Fax: +359 2 9768 311
Contact person: Atanas Kiryakov
E-mail: naso@sirma.bg

Vrije Universiteit Amsterdam (VUA)
Department of Computer Sciences
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
Tel: +31 20 444 7731, Fax: +31 84 221 4294
Contact person: Frank van Harmelen
E-mail: frank.van.harmelen@cs.vu.nl

Universitat Autonoma de Barcelona
Edifici B, Campus de la UAB
08193 Bellaterra (Cerdanyola del Vall` es)
Barcelona
Spain
Tel: +34 93 581 22 35, Fax: +34 93 581 29 88
Contact person: Pompeu Casanovas Romeu
E-mail: pompeu.casanovas@uab.es

Siemens Business Services GmbH & Co. OHG
Otto-Hahn-Ring 6
81739 Munich
Germany
Contact person: Dirk Ramhorst
Tel: +49 (89)63640225; Fax: +49 89 63640233
Email: Dirk.Ramhorst@siemens.com

Executive summary

This report presents the first release of the Massive Automatic Annotation component, developed on the basis of KIM Platform. After the description of the software architecture, it follows an overview of the different components – information extraction (IE), Knowledge base (KB), Indexing and Retrieval of semantic annotations (IR) and front ends. Performance and accuracy evaluations are made.

Most of the documentation is already available online and since there are constant changes in it, links are provided in this report instead of a copy. The documentation includes the precise steps for an installation, setup, and administration of the KIM Platform as well as the description of the APIs and the javadoc.

A motivation is presented for the development of a scalable semantic repository with light-weight reasoning capabilities, to be used by the massive automatic annotation component. A first version of such a semantic repository (called OWLIM) was already developed and its system documentation is given in Appendix A.

At the end, a KIM Fact Sheet is included in response to the SEKT guidelines for submission of software deliverables.

For the evaluation of KIM information extraction, we used three different corpora, each consisting of news articles in different domains: general international news, business news, and UK news. In order to combine the P/R metrics from the three different corpora, we used as a weight factor the number of tokens in each corpus divided by the total number of tokens for the three corpora. The achieved results are as follows:

Flat Named Entity Type	Precision	Recall	F1
Date	93.17%	93.63%	93.39%
Person	87.61%	90.87%	89.09%
Organization	82.29%	71.30%	76.03%
Location	92.77%	89.77%	91.23%
Percent	99.18%	97.69%	98.42%
Money	99.08%	98.72%	98.90%

KIM Performance. KIM is designed so as to ensure high accuracy and throughput within a robust architecture. Follow statistics about the scale and throughput on a basic ($1000-worth) PC:

- **Annotation speed: 10 kb/s**. The annotation speed depends primarily on the speed of the JAPE engine of GATE.
- **Indexing & Storage speed: 27 kb/s.** Based on Lucene.
- **Documents (with annotations) stored: 300,000**. Retrieval of a document by ID within a few milliseconds.

OWLIM Performance. The current version of OWLIM repository is capable of hanlding 30 million statements. It performs in-memory reasoning and query

answering, so, in order to add these 30M statements it needs 8GB of RAM. In order to evaluate its scalability a *City benchmark* experiment is performed on several machines with slightly different hardware configuration. The details and graphs can be found in Appendix A. The results can be summarized as follows:

- The **upload speed** (including inference and storage) varies in the range of 10,000-100,000 statements/second, depending on the machine and the size of the repository;
- The **maximum size of the repository** varies from couple million statements on a notebook to 30 million statements on a server with 8GB of RAM. As it can be expected, the 64-bit Java virtual machine requires slightly more memory for the same size of the repository;
- The **time for query evaluation** grows linearly with the size of the repository and the result count. It starts at tens of milliseconds, when the repository contains few millions of statements, and grows to few seconds when the repository gets bigger. Because all the query results are fetched, the total time for the query is affected by the size of the result, which grows linearly with the size of the repository, to reach tens of thousands of results;
- The **delete operation** is relatively slow, as it can be expected due to the straight forward invalidation of the inferred closure.

Unique achievements. Among the unique achievements in this deliverable it is worth to mention two:

- There was an independent KIM evaluation at the Semantic Annotation Workshop at ISWC-2005, where KIM was evaluated as the best automatic annotation platform [7];
- OWLIM is the fastest OWL repository, according to the limited evaluation data available (LUBM (50,0) benchmark) (see [10] for details).

Integration and application within SEKT. With respect to the integration and application in SEKT, KIM:

- Will be integrated to Ontology Based Information Extraction module (OBIE) – internal WP2 integration;
- Will be integrated to SIP (after the SIP V2 deliverable becomes available, a SIP annotator pipelet will be provided);
- Is used in the BT Digital Library case study for automatic annotation of documents and abstracts.

Note. This report subsumes a previous informal deliverable D2.6.0 submitted in October, 2005.

Contents

1 Overview

Fully-automatic metadata generation tools allow for annotating sets of documents as a batch job. The accuracy of the metadata generated this way will usually be lower than semi-automatic annotation tools, but there will be no human effort required. Tools for automatic annotation can also be trained on the metadata generated semi-automatically. These types of fully automatic process can have an important bootstrapping contribution for the application of knowledge technologies.

The annotations produced need to target the PROTON ontology [2]. Its general nature makes it suitable for fully automatic massive annotation, which, due to being cross-domain must be relatively simple. This generality can also make the task applicable in a lot of different contexts, wherever there is a large quantity of language data and a need to identify the basic elements present.

The Massive Automatic Annotation component is implemented on top of the KIM platform [1] which includes:

- PROTON ontology, KIMSO, KIMLO [3] and KIM World Knowledge base;
- KIM Server – with API for remote access and integration;
- Front-ends: KIM WebUI and a Plug-in for Internet Explorer.

The KIM Platform provides a novel Knowledge and Information Management (KIM) infrastructure and services for automatic semantic annotation, indexing, and retrieval of unstructured and semi-structured content.

As a base line, KIM analyzes texts and recognizes references to entities (such as persons, organizations, locations, dates). Then it tries to match the reference with a known entity that has a unique URI and description. Alternatively, a new URI and description are generated automatically. Finally, the reference in the document is annotated with the URI of the entity. We call this process (as well as its result) a *semantic annotation* (figure 1). This sort of meta-data can be used for indexing, retrieval, visualization and automatic hyper-linking of documents.

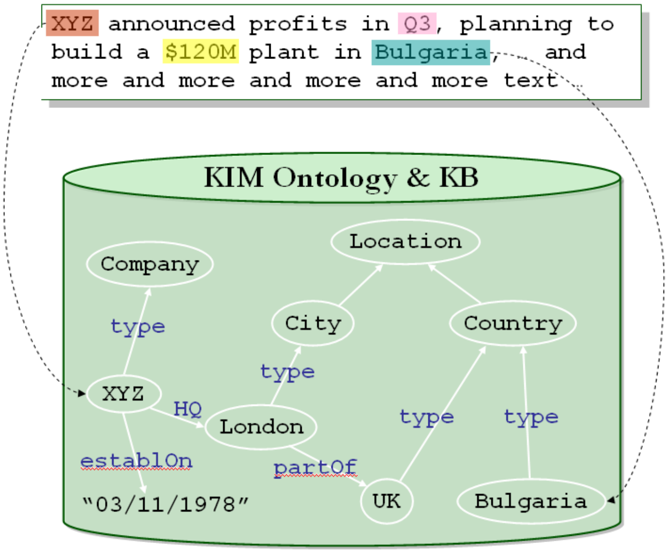

XYZ announced profits in Q3, planning to build a $120M plant in Bulgaria, and more and more and more and more text
KIM Ontology & KB
Location
Company
type
City
Country
type
XYZ
HQ
London
type
type
establOn
partOf
"03/11/1978"
UK
Bulgaria

Figure 1. Semantic annotation

In order to enable the easy bootstrapping of applications, KIM is based on PROTON ontology, which consists of about 250 classes and 100 properties. Furthermore, a knowledge base (KIM KB), pre-populated with about 200, 000 entity descriptions, is bundled with KIM. Its role is to provide as a background knowledge (resembling a human's common culture) a quasi-exhaustive coverage of the entities of general importance - those, which are considered well-known and thus not explicitly introduced in the documents. For this reason, it is hard to extract automatically their descriptions.

From a technical point of view, the architecture allows KIM-based applications to perform automatic semantic annotation, content retrieval, based on semantic restrictions, as well as querying and modifying of the underlying ontologies and knowledge bases.

An extended overview of the semantic annotation approach taken in KIM can be found in [8]. This paper contains also extensive analysis on the state-of-the-art in the area of automatic semantic annotation tools. [8] as well as most of this document presents the basics of KIM. An extensive list of its multiple new features and releases can be found online on http://www.ontotext.com/kim/release-notes.html A recent evaluation of automatic semantic annotation tools reported in [7] has outlined KIM as the system with the best overall performance; the evaluation had included most of the state-of-the-art systems.

2 KIM Architecture

The KIM platform consists of formal knowledge resources (KIM Ontology[1], world knowledge base), KIM Server (with API for remote access, embedding, and integration), and front-ends. The architecture of KIM Server (figure 2) allows easy

[1] PROTON + KIMSO and KIMLO extensions

modification, extension, and embedding in third-party systems. It also provides an abstraction layer over the specific underlying component implementations, and thus ensures flexibility in case that custom implementation (or configuration) of KIM with another semantic repository, metadata storage or IR engine is made. Furthermore, KIM Server components could easily be wrapped in the shape expected by another component-based framework, an approach that minimizes the integration costs. KIM Server has the following major components: Semantic Repository, Semantic Annotation, Document Persistence, Indexing and Query. They are visible as parts of the KIM Server API and could be used by third-party systems/applications.

The KIM platform is based on robust open-source platforms specialized in three different domains: OWL DLP repositories, HLT (and especially IE), and IR. The technologies on which KIM is built have been carefully chosen, so that they satisfy, among other, the following conditions: to be mature enough, scalable and platform independent. The knowledge resources are kept in the Sesame-based[2] OWLIM[3] repository, which provides storage and query functionality infrastructure. The KIM Architecture is compatible with any Sesame-based repository, but for massive automatic annotation it requires the speed and the scale that currently is offered only by OWLIM. It was queried by means of the semantic search methods to identify the entities in accordance to predefined restrictions, and the result is used for the retrieval of the referring documents. For its initialization and further processing, the IE process also relies on the semantic repository.

The GATE [4] platform has been used as a basis for the IE process and also for the management of content and annotations. It provides the fundamental text analysis technologies, on which we have built the semantically aware extensions, specific for the IE of KIM. The annotations and document management paradigms have been derived from the GATE infrastructure, though slightly simplified in order to avoid KIM clients to depend on anything besides the KIM API.

The Lucene [5] IR engine has been adopted to perform indexing and retrieval with respect to named entities and evaluation of content relevance according to the entities. This allows the semantic access methods described in section 7. The adoption of Lucene proves that it is easy to adjust a traditional IR engine to perform indexing with respect to metadata.

[2] Sesame, http://www.openrdf.org, is a RDF(S)/OWL repository developed by Aduna BV

[3] OWLIM (Appendix A to this document) is a light-weight fast and scalable OWL-DLP repository.

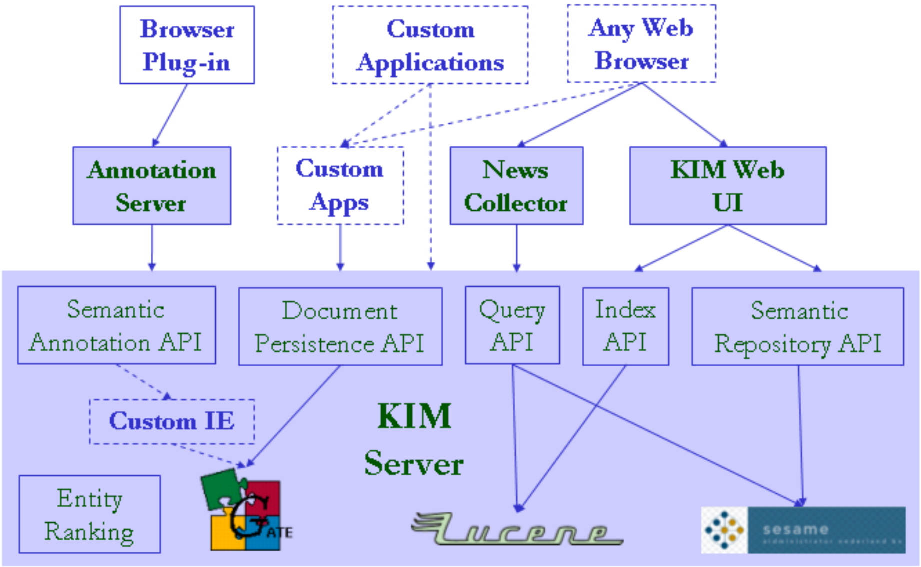


Figure 2. KIM Platform Architecture

2.1 KIM World Knowledge Base (KB)

KIM KB has been pre-populated with entities of general importance, which empower the IE process to perform well on inter-domain document content. Because the building of a domain-independent general knowledge base is a complex task and, defined in this way, it does not point to an obvious realization strategy, we substituted the task with an easier one, which seems to serve as a good approximation: to build a KB that provides a good coverage of the entities mentioned in the international news. Here we mean those publications that cross the borders of the countries and feed the headlines of the global news wires. The specifics of such a domain is that it covers (and also pre-determines) the most well known entities in the world. At the base line, entity descriptions include entities with their proper classes and aliases, but various entity relations and attributes are also predefined (like the position of a person within an organization or the location of a company.)

2.2 KIM KB Prepopulation

KIM KB has been pre-populated with entities of general importance, which give enough clues for the IE process to perform well on inter-domain web content. It consists of more than 200,000 entities.

At its current state, the KIM KB contains about 36,000 locations, including continents, global regions, countries (according to FIPS) with their capitals, 4,400 cities (including all the cities with a population over 100,000), mountains, big rivers, oceans, seas, and even oil fields. Each location has geographic coordinates and several aliases (usually including English, French, Spanish, and sometimes the local transcription of the location name) as well as co-positioning relations (e.g. `subRegionOf`.) This spatial knowledge provides a good basis for location-based services.

The organizations with high general importance also have been pre-populated in the KB. Including the biggest world organizations (such as UN, NATO, OPEC), some of the Semantic Web research related organizations (both academic and commercial), over 140,000 international companies, and 140 stock exchanges, for a total of 147,000 organization instances. For some of the public companies there are position relations of their managerial personnel. The organizations also have **`locatedIn`** relations to the corresponding **`Country`** instances. The additionally imported information about the companies consists of short description, URL, reference to an industry sector, reported sales, net income, and number of employees.

Finally, in order to enable the IE process to recognize new entities and relations not part of the KB, a collection of lexical resources was also added to the KB. It covers organization suffixes, names of persons, time lexica, currency prefixes and others.

In addition to the KB described above, we produced also a smaller version, which is a step towards less restrictive distribution constraints – both in terms of licensing and as regards the hardware requirements. Of course, this was achieved by removing some of the entities, but we verified that the removal has no serious impact on the IE's accuracy. We expect that the small version will thus be more usable as a basis for domain-specific extensions. Table 1 shows a comparison between the two versions.

Instances	Small KB	Full KB
- Entity:	40,804	205,287
- Location:	12,528	35,590
- Country:	261	261
- Province:	4,262	4,262
- City:	4,400	4,417
- Organization:	8,339	146,969
- Company:	7,848	146,262
- Person:	6,022	6,354
- Alias:	64,589	429,035

Table 1. Statistics about *full* and *small* versions of the KB.

2.3 Controlling the Quality and Coverage of KIM KB

To ensure the quality of the KB content, is not a trivial task and it is not possible to be performed manually (with more than 200,000 pre-populated entities, the manual approach simply does not scale). The KIM KB is iteratively verified using an independently built *Test KB* of entities and relations collected manually from various web sources. During the evaluation of the performance of the KIM IE against a human annotated corpus, an indirect verification was also performed.

The coverage of the KIM KB is guaranteed by means of processing and analysis of the leading articles of the global news wires. The corpus of these articles is updated constantly and enriched with approximately 4000 documents each week – using the top stories, as well as all the main economic and political news, collected from about

15 sources. On top of the corpus gathered this way, entity ranking is performed so as to detect the level of “popularity” that specific entities possess. This approach allows at least the proper manual handling of the most popular entities, as well as the early spotting of problems with the import strategy and sources. The ranking algorithm works on entities and treats all of the entity’s aliases as equivalent references. Because of that, the algorithm is very sensitive to duplicated instances (e.g. when two aliases are presented as two separate instances). We faced a number of issues, related to the identification of instances. Examples range from problems arising when using numbers or stop-words for Locations to problems arising when using variations in the punctuation as well as suffixes for Organizations (e.g. “The Coca Cola Company” and “Coca-Cola” are two aliases of the same entity). To ensure a consistent pre-population of the KB, we used some heuristics, applied in different combinations, depending on our “trust” in the source, including:

- suppressing of aliases according to various criteria – secondary aliases matching primary alias of another entity; word lists: a stop-word list, a list of common words, a list of about 80,000 English words;
- class-specific pre-processing and comparison of aliases;
- automatic generation of additional aliases: e.g. by truncating parts of the main alias - if “Xyz Ltd.” is the main alias then we might expect that “Xyz” is also a relevant alias.

3 Information extraction architecture

As it was already mentioned, KIM IE is based on the GATE framework, which has proved its maturity, extensibility and task independency for IE and other NLP applications. The essence of the KIM IE lies in the recognition of named entities with respect to the PROTON 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 KB. For new (previously unknown) entities, new identifiers are allocated and assigned; then minimal descriptions are stored in the semantic repository. The annotations are kept separated from the annotated content, and an API for their management is provided.

The default KIM IE application is based on semantic gazetteers, shallow analysis of the text, and pattern-matching grammars. The evaluation has been performed with respect to flat NE types (e.g. if Reuters is recognized as **`NewsAgency`**, on a more general level it is an **`Organization`**.) We evaluate against corpora of flat NE types (all among PROTON Classes from the PROTON-Top layer).

For the evaluation of KIM IE, we used three different corpora, each consisting of news articles in different domains: general international news, business news, and UK news. The results achieved are presented in Table 2. In order to combine the P/R metrics from the three different corpora, we used as a weight factor the number of tokens in each corpus divided by the total number of tokens for the three corpora.

Table 2. KIM IE Evaluation

Flat NE Type	Precision	Recall	F1
Date	93.17%	93.63%	93.39%

Person	87.61%	90.87%	89.09%
Organization	82.29%	71.30%	76.03%
Location	92.77%	89.77%	91.23%
Percent	99.18%	97.69%	98.42%
Money	99.08%	98.72%	98.90%

The task of creating a Semantic IE application benefited from the existing IE components in GATE, but we had to enable some of them semantically (e.g. the pattern-matching transducer), or to create completely new components – such as the semantic gazetteer.

Another important issue is the extensibility as well as the opportunity to change completely the IE application used in KIM. Any GATE application (IE pipe-line) could be plugged into the KIM Server. It could include machine learning, as well as rule-based components (or an arbitrary set of the palette of NLP components integrated in GATE). The IE application could also be provided by a completely independent system, if it was appropriately wrapped and plugged into KIM.

A substantial difference of the semantic IE process as compared to the traditional one is the fact that it is not only able to find out the (most specific) type of the extracted entity, but also to *identify* it, by linking the entity to its semantic description in the instance base. This approach makes possible entities to be traced across documents and their descriptions to be enriched through the IE process.

The full explanation and discussion of how the IE components were semantically enabled can be found in [9]. Here we will present the IE component flow diagram (Figure 3. Semantic IE architecture) that displays the sequential processing of content to the point where semantic annotations of NE are produced over it. The semantic repository is also displayed and linked with the ontology and KB aware components. The semantically-aware modules are presented as subsections below:

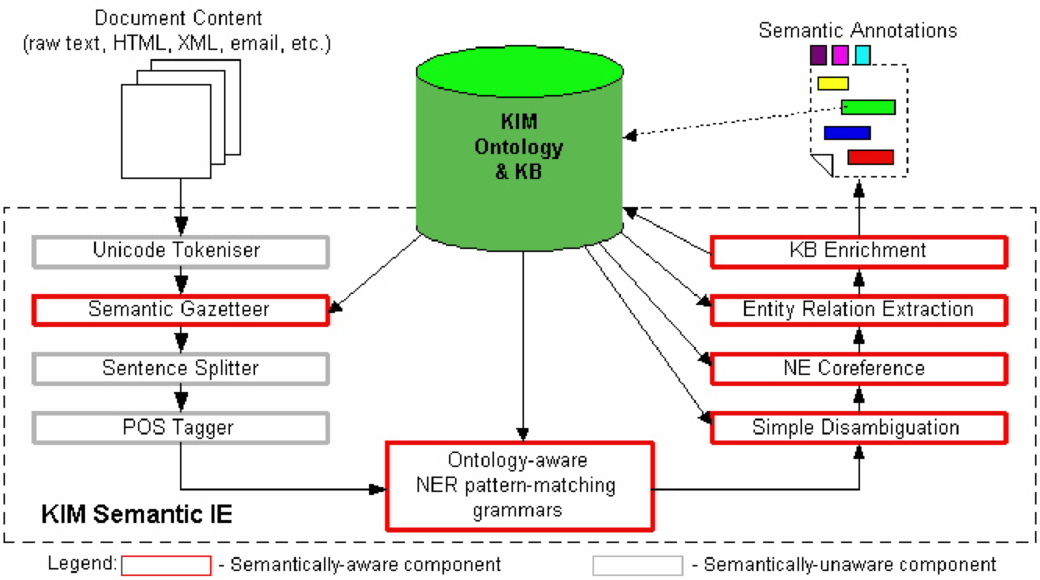


Figure 3. Semantic IE architecture

3.1 Semantic Gazetteer

The lists of a traditional text-lookup component have been exchanged with a knowledge base that keeps the entities with their aliases and descriptions, as well as the lexical resources (such as possible male person first names). These are used to initialize the semantic gazetteer component, which keeps the various aliases and their type and instance references (URIs). Upon occurrence of a known lexical resource or entity alias in the text (f.e. Monday, John, GMT, etc.), the semantic gazetteer generates a temporal annotation with a link to a class in the ontology (f.e. Monday will be linked to the PROTON ontology class *DayOfWeek*). What is more, the aliases of entities in the text are linked to the specific instances they refer to (e.g. California will be linked to the instance *Province.4188*).

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

Although the KB contains both pre-populated and automatically recognized entities, only the former are used in the lookup process. The entities extracted from the processed content are not considered, and thus possible recognition mistakes are not reused as evidences.

3.2 Ontology-Aware Pattern-Matching Grammars

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

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

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

3.3 Orthographic NE Coreference

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

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

3.4 Simple Disambiguation

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

Another class of ambiguities is the appearance of two annotations with different class and instance information over the same entity reference (New York being a Province and a City). Currently disambiguation of such annotations is not performed and this is subject of future work. For example, the context could be scanned for entities related to the ambiguous ones and thus relevance of the alternative entities to the content could be evaluated. For instance, if Moscow is used along with Russia its relevance is higher than the relevance of the alternative american city. We would experiment with techniques similar to those used for word-sense disambiguation (namely, lexical-chaining) and “symbolic” context management.

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

3.5 KB Enrichment

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

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

This finalizes the IE process, having as a result named entity annotations linked to their semantic descriptions in the KB. We call this particular IE process "semantic annotation". Two sorts of metadata are generated in this process:

- The descriptions of the newly generated entities (see KB enrichment above). It is a matter of terminology, whether these descriptions represent metadata or ontology population.
- The generation annotations represent metadata about the documents.

4 Indexing and Retrieval

KIM provides indexing with respect to the semantic annotations generated for a document, i.e. indexing with respect to the metadata. This type of indexing enables new (semantically-enhanced) access methods (or user-need definitions.) Thus, the user could specify queries that consist of constraints about the types of the entities, relations between the entities, and entity's attributes. This is, one could specify the NEs to be referred to in the documents of interest, using name restrictions (e.g. a **`Person`** which name ends with 'Alabama'). An example of a query consisting of pattern restrictions over entities is as follows:

- give me all documents referring a **`Person`** that **`hasPosition`** "CEO" within a **`Company`**, **`locatedIn`** a **`Location`** with name "UK".

To answer the query, KIM applies the semantic restrictions over the entities in the instance base. The resulting set of entities is matched against the index, produced by the semantic indexing of the processed documents. Then the referring documents are retrieved with relevance ranking according to these NEs. Such queries could also be combined with traditional keyword search and thus, could benefit from the combination of both approaches (e.g. via intersection or union). Technically, the Lucene IR engine is adapted to perform full-text indexing, uniquely addressing each entity and disregarding the alias used in the text.

The retrieval accuracy of KIM has not been evaluated against a traditional IR engine, a topic that should be addressed in the future. However, KIM has the potential to perform better, not only towards reducing the unrelated documents in the result set while still retrieving the relevant ones, but also towards an increase in the number of the relevant documents by those that do not contain the alias, used for the entity name restriction, but which nonetheless contain the same entity, mentioned with another of its aliases. For example, if you look for documents that refer to the city of "Beijing" and you use a keyword search specifying the city by its name, then you will miss all documents that mention only "Pekin" ("Beijing" and "Pekin" are two aliases of the capital of China; the Bejing being the main alias). On the contrary, given the world knowledge in KIM, the semantic IR would also find the documents that only mention "Pekin", because (i) the KB knows that Pekin and Beijing are aliases of the same entity and (ii) the documents are indexed by the entity identifier. Name abbreviations and their full forms can serve as another example.

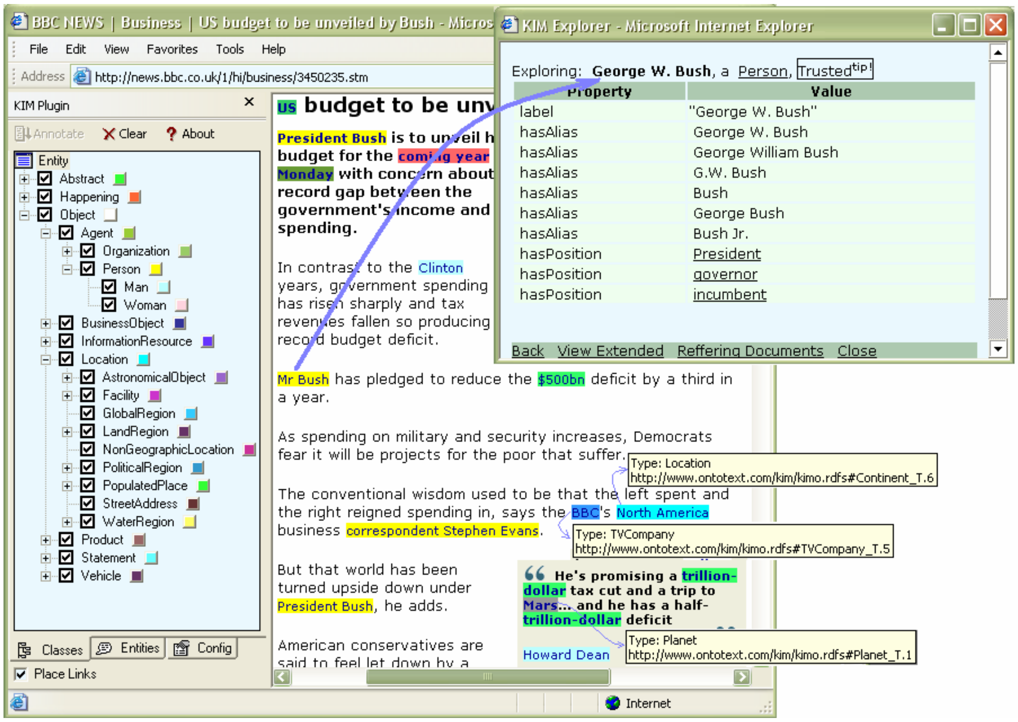
The IR functionality is available through the API and through the KIM Web UI. The API allows the creation of semantic queries, and requests the documents, that refer to the restrictions, from a particular data-store. As a result some of the features (title, author, origin, etc.) of the resulting documents are loaded from the data-store, but the documents are not loaded completely in order to delay expensive processing. The same functionality is made available through the Web.

Currently the documents in the result are not ordered by their relevance. Relevance ranking can be based on the number of entities in the document that match the query. The rationale behind this decision not to implement ranking are present is twofold:

- there will be a performance penalty for implementing such ranking because currently we use the underlying IR engine (Lucene) to query the documents without actually retrieving their content;
- we are working on a CoreDB component (see Section 9 Future plans) that is likely to replace the whole IR subsystem and also to provide different means of ranking (e.g. ranking of documents based on the global ranking of entities that occur in them).

5 KIM Front Ends

The KIM Server API provides the possibility to build different front-end user interfaces. These front-ends could provide full access to the functionality of the KIM Server, including its IR functionality, semantic repositories, semantic annotation services, and document and metadata management infrastructure. Some front-ends have already been built-in in the KIM Platform. These are the browser plug-in, the Web UI, and KB Explorer.



Exploring: **George W. Bush**, a Person, Trusted tip!

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

US budget to be unv

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

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

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

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

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

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

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

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

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

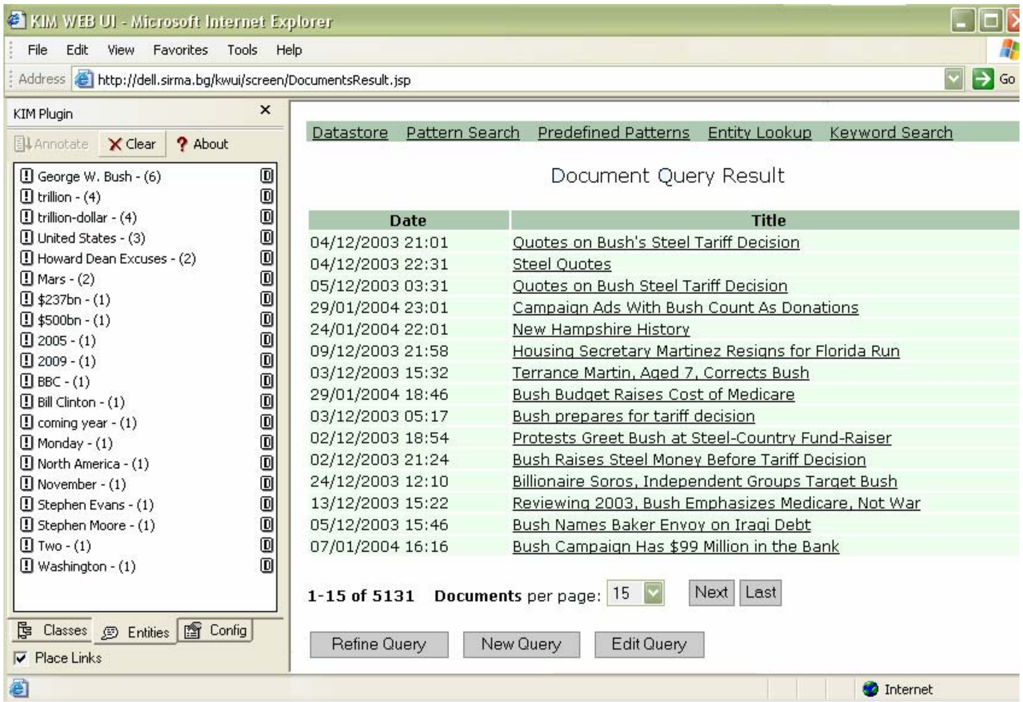
Howard Dean

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

American conservatives are said to feel let down by a

Figure 4. KIM Internet Explorer Plug-in

We have created a plug-in (Figure 4. KIM Internet Explorer Plug-in) for the MS Internet Explorer browser. The KIM plug-in provides light-weight delivery of semantic annotations to the end user. On its first tab, the plug-in displays the entity type branch of the PROTON ontology. For each entity type there is an associated color used to highlight 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 (arbitrary) browser content, the plug-in extracts the text of the currently displayed document and sends it to an Annotation Server which in its turn uses the KIM Server Semantic Annotation 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 hyperlinked to the KIM KB Explorer. The tooltips on the bottom right contain the *type* and *unique identifier*, visible when the cursor is positioned on an annotation.



http://dell.sirma.bg/kwui/screen/DocumentsResult.jsp

- George W. Bush - (6)
- trillion - (4)
- trillion-dollar - (4)
- United States - (3)
- Howard Dean Excuses - (2)
- Mars - (2)
- $237bn - (1)
- $500bn - (1)
- 2005 - (1)
- 2009 - (1)
- BBC - (1)
- Bill Clinton - (1)
- coming year - (1)
- Monday - (1)
- North America - (1)
- November - (1)
- Stephen Evans - (1)
- Stephen Moore - (1)
- Two - (1)
- Washington - (1)

☑ Place Links

Document Query Result

Date	Title
04/12/2003 21:01	Quotes on Bush's Steel Tariff Decision
04/12/2003 22:31	Steel Quotes
05/12/2003 03:31	Quotes on Bush Steel Tariff Decision
29/01/2004 23:01	Campaign Ads With Bush Count As Donations
24/01/2004 22:01	New Hampshire History
09/12/2003 21:58	Housing Secretary Martinez Resigns for Florida Run
03/12/2003 15:32	Terrance Martin, Aged 7, Corrects Bush
29/01/2004 18:46	Bush Budget Raises Cost of Medicare
03/12/2003 05:17	Bush prepares for tariff decision
02/12/2003 18:54	Protests Greet Bush at Steel-Country Fund-Raiser
02/12/2003 21:24	Bush Raises Steel Money Before Tariff Decision
24/12/2003 12:10	Billionaire Soros, Independent Groups Target Bush
13/12/2003 15:22	Reviewing 2003, Bush Emphasizes Medicare, Not War
05/12/2003 15:46	Bush Names Baker Envoy on Iraqi Debt
07/01/2004 16:16	Bush Campaign Has $99 Million in the Bank

1-15 of 5131 **Documents** per page: 15 Next Last

Refine Query New Query Edit Query

Figure 5. The *Entities* tab of the plug-in with the referring documents for one of the entities

The second tab (as shown on Figure 5. The *Entities* tab of the plug-in with the referring documents for one of the entities) of the plug-in contains a list of all the entities recognized in the current document, sorted by frequency of their appearance (the most frequent - on the top). By following the link that is placed over an identified entity in the content, or by choosing from the list of entities, the user invokes KIM KB Explorer to display the entity's semantic description in the KB (incl. type, aliases, relations and attributes). In this way, the user can navigate directly from the mentioning of entities in the text to their linked instances in the KB.

The second tab contains also the option (by clicking on a little icon) to execute a semantic IR request to the *default* (for the KIM Server used by this plug-in) document data-store. The result is a list of all the documents that mention this particular entity. To gain an impression of the usefulness of the above, consider how one finds a page about a particular organization, while browsing and annotating, and goes to the Entities tab of the plug-in, requests the referring documents and gets a result list, that could be explored further.

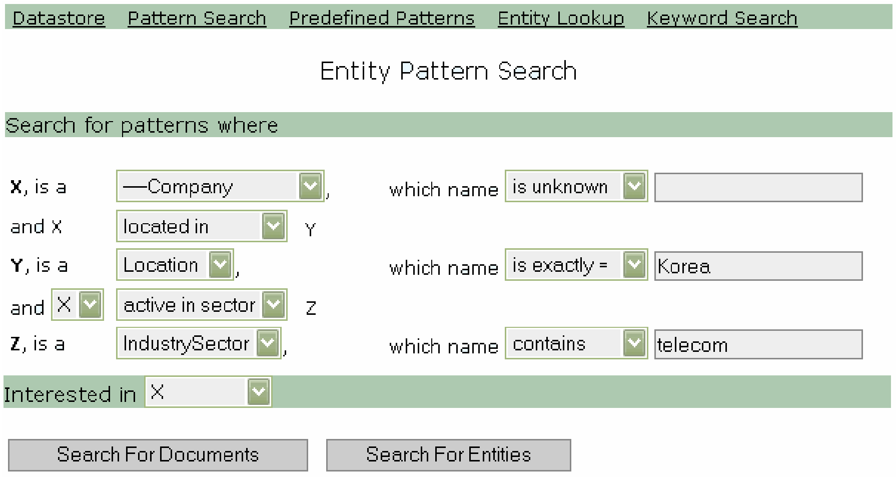
Another front-end is the KIM Web UI, which offers IR services over data-stores of semantically annotated and indexed documents. It offers three types of semantic queries that can be seen as three levels of complexity of the semantic queries:

- *entity lookup* – search for an entity by alias and type
- *entity pattern search* – a search for a scenario with participants: entities, relations between them, restrictions by classes, aliases and attributes
- *predefined patterns* – a simplification of the above search, by pre-selecting some of the choices.

All these queries could be combined with the traditional keyword search, also available in the Web UI along with typical metadata properties for documents such as authors, title, subtitle, and subject. Any combination of queries could result in a set of entities that satisfy the restrictions, or in a set of documents that refer to these entities.

The *entity lookup* allows restrictions by the type and alias of the entity. E.g. give me all organizations that end on "Ltd". The *predefined patterns* search provides a set of frequently used queries to assist the user. These queries consist of a predefined pattern frame of entities with specified types (like **`Person - hasPosition - Position - within Organization`**). The user is allowed to restrict the entities in the pattern by the enitity's name (e.g. "CEO" for the **`Position`**.) The most comprehensive query definition interface provided is the *entity pattern search*. It has the flexibility to specify the entity types, relations between these entities, and thus to create the entity pattern. Furthermore, one could specify attribute restrictions (such as **`alias, longitude, age`**, etc.)

After the specification of the user need the next step is to retrieve the relevant entities or documents (referring to those entities). The result set could be further narrowed by refining the queries. The content of the documents from the result set could be examined. The mentions of the relevant entities are highlighted and hyperlinked to the KB Explorer. If the result of the query is a set of entities – they are linked to the KB Explorer, which is capable to display their semantic descriptions in the instance base.



Entity Pattern Search

Search for patterns where

X, is a —Company, which name is unknown

and X located in Y

Y, is a Location, which name is exactly = Korea

and X active in sector Z

Z, is a IndustrySector, which name contains telecom

Interested in X

Figure 6. Entity Pattern Search from the Web UI - looking for a telecom company in Korea

5.1 KIM Document Store Population Tool

The document store population tool is a standalone tool, that can be run on the same machine where the KIM Platform is installed. It allows documents from a directory (and its subdirectories) to be quickly added to the KIM Document Store.

A screenshot and the latest documentation can be found at:

http://www.ontotext.com/kim/doc/sys-doc/kim-platform-administration/populate-doc-store.html.

6 Performance

KIM is designed so as to ensure high accuracy and throughput within a robust architecture. Scale and throughput on a $1000-worth PC:

- **Annotation speed: 10 kb/s**. The annotation speed depends primarily on the speed of the JAPE engine of GATE.
- **Indexing & Storage speed: 27 kb/s.** Based on Lucene.
- **Documents (with annotations) stored: 300,000**. Retrieval of a document by ID within a few milliseconds.

7 OWLIM – a light-weight high-performance semantic repository

The Massive Automatic Annotation produces lots of metadata, while at the same time it does not depend on heavy reasonning infrastructure. Thus, it needs a scalable repository, capable of handling tens of millions statements with OWL DLP support.

We have developed a first version of such a repository (called OWLIM) which is capable of handling 30 million statements. It scored best on LUBM(50,0) benchmark and this makes it (with respect to LUBM benchmark) the fastest OWL DLP repository currently available. It will be further improved in the course of the next year and will be available as a separate formal deliverable to all SEKT partners at M36.

The full OWLIM System documentation is available as *Appendix A – OWLIM System Documentation*.

8 Application of Massive Automatic Annotation Task within SEKT

Massive Automatic Annotation task is used in BT Digital Library case study, where documents and abstracts are being annotated automatically with respect to PROTON ontology. Further, the KIM platform is used in WP5 Search & Browse tool.

Currently the automatic annotation is performed via the KIM IE modules which were tuned to work in general (not domain-specific) context. In order to allow automatic annotation for specific domain (such as BTDL content) and annotations to be performed with respect to another ontology (such as BTDL ontology) KIM needs to be integrated to the OBIE component (T2.1). The integration will allow an already trained component to be used for the IE as part of the semantic annotation. The integration will be implemented on the basis of the GATE framework (not through SIP) because:

- The KIM IE process is already based on GATE, so, the integration of an OBIE component will be very efficient and straightforeward;
- Any delay caused by loose integration in the main processing pipeline directly affects the performance of the whole massive automatic annotation process.

SIP integration:

- In order to allow modules from other technical workpackages and case study applications to use massive automatic annotation, we will provide KIM Annotator Pipelet as an external component (stateful) available through SIP by M27;
- For more advanced usage of the KIM platform, the KIM API is available, which is extensively documented on its own. At present there is no need to write SIP adaptors for the whole API, because the API is intended for software engineers and not for SIP application developers. However, it will be interesting to investigate the possibility of providing an uniform access to KIM from within SIP through something like JNDI, so a client application can ask SIP "give me the automatic annotation component" and then can use its API.

9 Future plans

In order that the full massive annotation functionality is used in SEKT, this functionality should be delivered earlier than the next formal deliverable (which is at M36). Therefore we will provide an informal deliverable of KIM platform at M30, which will contain:

- KIM Cluster architecture, which enables several annotators to work in a cluster with a centralized ontology repository and document store;
- CoreDB and Timelines – Co-Occurrence and Ranking of Entities (CORE) DB will be a new component of the KIM platform which allows to ask in a structured manner for:
    - The number of references to entities in a (sub-)set of documents;
    - The entities, which co-occur together with other entities.
- Integration of KIM with KAON2 (through PROTON API – D5.0.2) for use in BTDL case study;
- OBIE integration.

The objectives for the next formal deliverables at M36 are:

- D2.6.2 – deliver a finalized version of the functionality for M30 and scale to 5M documents with respect to BTDL case study;
- D2.6.3 (OWLIM formal deliverable) - further development of OWLIM: add file-based storage for increased scalability.

10 KIM Fact Sheet

Following the SEKT guidelines for submission of software deliverables we list the appropriate facts in this section.

10.1 Documentation

10.1.1 APIs

The full documentation of the KIM API can be found at:

http://www.ontotext.com/kim/doc/sys-doc/index.html

10.2 Required operating system / environment

Currently supported operating systems are: Windows, Linux and Solaris. KIM is written in Java and to that extent it is independent from the operating system. The system requirements are available as part of the system documentation:

http://www.ontotext.com/kim/doc/sys-doc/kim-platform-administration/kim-platform-setup.html

10.3 Licence

KIM Platform Licence Agreement is described at:

http://www.ontotext.com/kim/KIM-licence-agreement.html

The licence also includes the licence terms of the third party software used in KIM Platform.

10.4 Downloads and installation instructions

The release notes for the latest version are available at:

http://www.ontotext.com/kim/release-notes.html

The installation of the *KIM Platform* is available from

http://www.ontotext.com/kim/KIM-downloads.html

You will need to register as KIM user (free) before downloading the installation. The system requirements and the installation instructions are available as part from the system documentation:

http://www.ontotext.com/kim/doc/sys-doc/kim-platform-administration/kim-platform-setup.html

10.5 Other requirements of the software

KIM uses the following third party software: GATE, Sesame, Lucene, Touch-Graph and Ontology Middleware Module.

For full references to these products, please refer to the KIM Platform Licence, section "TERMS AND CONDITIONS RELATING TO THIRD-PARTY PRODUCTS AND LIBRARIES". These third party products do not require separate installation.

Optionally the installation installs KIM WebUI as a web application for Apache Tomcat [3], which needs to be installed separately in the system.

10.6 Unit/component testing

Different techniques are used for different components in the KIM Platform:

- Information extraction: There is a regression testing tool, which evaluates the IE accuracy (in terms of precision and recall) against 3 different corpora and produces a comparison with the previous run (see section 3 Information extraction architecture)
- KIM KB Quality: is evaluated against a smaller independently built *Test KB* (see section 2.3 Controlling the Quality and Coverage of KIM KB)
- Unit-level (java): There are automated unit tests (JUnit) for the different KIM components

11 Bibliography and references

[1] KIM, http://www.ontotext.com/kim

[2] PROTON and PROTON-KM description, D1.8.1 Base Upper Level Ontology Guidance

[3] KIMSO – KIM System Ontology. KIMLO – KIM Lexical Ontology. Both are KIM-specific PROTON extensions.

[4] GATE, http://www.gate.ac.uk/, General Architecture for Text Engineering, developed by the NLP group at University of Sheffield

[5] http://lucene.apache.org

[6] http://jakarta.apache.org/tomcat

[7] Peyman Sazedj and H. Sofia Pinto, **Time to evaluate: Targeting Annotation Tools**. In the Proc. of *Semannot 2005 Workshop*, ISWC 2005, Nov 2005.

[8] Kiryakov et al, **Semantic Annotation, Indexing, and Retrieval.** Elsevier's Journal of Web Sematics, Vol. 2, Issue (1), 2005.

[9] Borislav Popov et al, **Towards Semantic Web Information Extraction** Human Language Technologies Workshop at the 2nd International Semantic Web Conference (ISWC2003), 20 October 2003, Florida, USA.

[10] Kiryakov et Al. **OWLIM – a Pragmatic Semantic Repository for OWL** In Proc. of Int. Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2005), WISE 2005, 20 Nov, New York City, USA.