Combining and Uniting Business Intelligence with Semantic Technologies

Acronym: CUBIST
Project No: 257403

Small or Medium-scale Focused Research Project
FP7-ICT-2009-5
Duration: 2010/10/01-2013/09/30

Semantic ETL from unstructured data sources Report v.1

Abstract:

Type:	Report
Document ID:	CUBIST D2.1.1
Workpackage:	WP2
Leading partner:	ONTO
Author(s):	Marin Dimitrov (ONTO) Alex Simov (ONTO)
Dissemination level:	PU
Status:	final
Date:	18 November 2011
Version:	1.0

Versioning and contribution history

Version	Description	Contributors
0.1	Initial version	Alex Simov
0.2	Final version	Marin Dimitrov
0.3	Proof reading & corrections	Marin Dimitrov
0.4	Updates according to Robert Rieger (SAP) review comments	Alex Simov
1.0	Emre Sevinç (SAS) review notes implementation	Alex Simov

Table of contents

1 Introduction

The goal of this deliverable is to provide a summary of existing approaches towards extracting information from unstructured datasources, existing text mining and semantic annotation platforms that can be adapted and deployed as part of the Data Integration and Federation Platform of CUBIST (D2.3.1), as well as an assessment of the suitability of these platforms with respect to the particular requirements of CUBIST use cases for analyzing unstructured data.

The three use cases of CUBIST vary significantly with respect to their requirements for extracting information from text. On the one side of the spectrum is the HWU use case, where there are no unstructured data sources at present, so no text mining and extraction is required at all in the context of the use case. The SAS use case provides a mixture of mostly structured and some unstructured data sources, where the expectation is to be able to identify entities and relations from texts and interlink them with the corresponding documents. The INN use case is the one where the text mining task is of highest importance, not only because most of the data sources are unstructured, but also due to specific requirements of the use case for automated qualification extraction and sentiment/opinion mining, which go beyond the traditional information extraction and semantic annotation.

This deliverable is organised as follows:

- Section 2 provides a brief introduction to the domain of text mining and semantic annotation which will be used in order to extract and integrate information from unstructured data sources.
- Section 3 provides a brief overview of the GATE and KIM text mining and semantic annotation platforms which will be used and extended within D2.3.1 in order to support the requirements of the CUBIST use cases.
- Finally, section 4 provides an assessment of the suitability of the GATE and KIM platforms with respect to the specific requirements of CUBIST use cases with respect to extracting information from unstructured data sources.

2 Text mining

Text mining is the process of deriving high-quality structured information from unstructured (textual) data in natural language. This involves the process of:

a) structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database);

b) deriving patterns within the structured data;

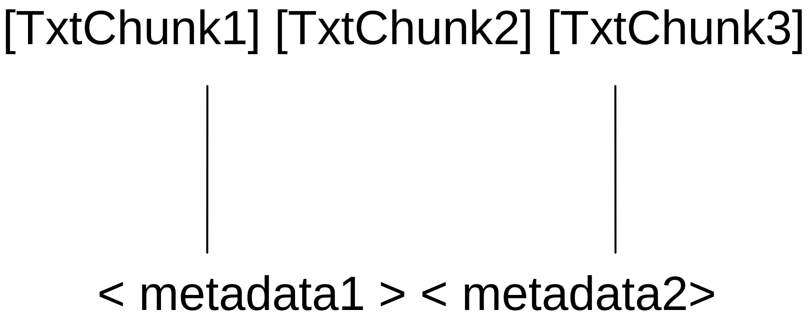
c) evaluation and interpretation of the output.

Typical text mining tasks include:

- text categorization,
- concept/entity extraction,
- taxonomy or ontology generation,
- sentiment/opinion analysis,
- document summarization,
- relationship extraction (*i.e.*, learning relations between named entities).

The textual datasources – for example: user manuals, instructions, reports, etc – contain information which has to be processed and transformed into structured metadata in order to be aligned with existing entities in the knowledge base (semantic database). The generation of metadata is a process of information extraction and assigning information based on a given ontology in the form of annotations.

The acquiring of metadata from texts happens with natural language processing (NLP) based techniques which identify relevant pieces of information. Usually natural language processing techniques are being used in the language expression identification phase. In this phase chunks of text are identified as describing information referred to with certain metadata (Figure 1).


[TxtChunk1] [TxtChunk2] [TxtChunk3]

< metadata1 > < metadata2>

Figure 1. Transformation of text into metadata

The transformation of textual information into metadata is the process of identifying chunks of text, sequences of words, as describing particular metadata.

With the rapid growth of electronic resources and the continuous emergence of new information requirements, acquiring effective metadata has become more difficult. Major problem with processing the content in order to transform it into metadata is the recognition of the right textual elements and mapping them to the correct elements of the existing knowledge base, e.g. finding one representation of a unique object.

Typical challenges that arise during the process of text mining are:

- How to identify relevant information
- How to aggregate relevant information
- How to select relevant information
- How to present relevant information

2.1 Semantic Annotation

Semantic annotation is the process of identifying knowledge elements in text and mapping them to instances and entities in a given knowledge base. It is the process of automatic generation of named-entity[1] annotations with class and instance references to a semantic repository (Figure 2). Semantic annotation is applicable for web, non-web documents, and text fields in databases.

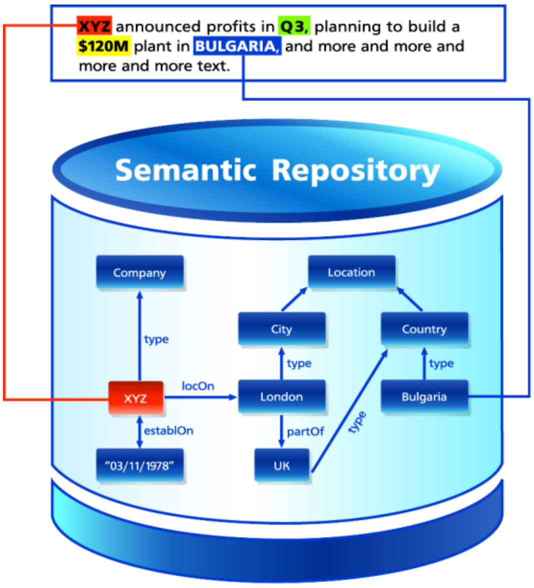
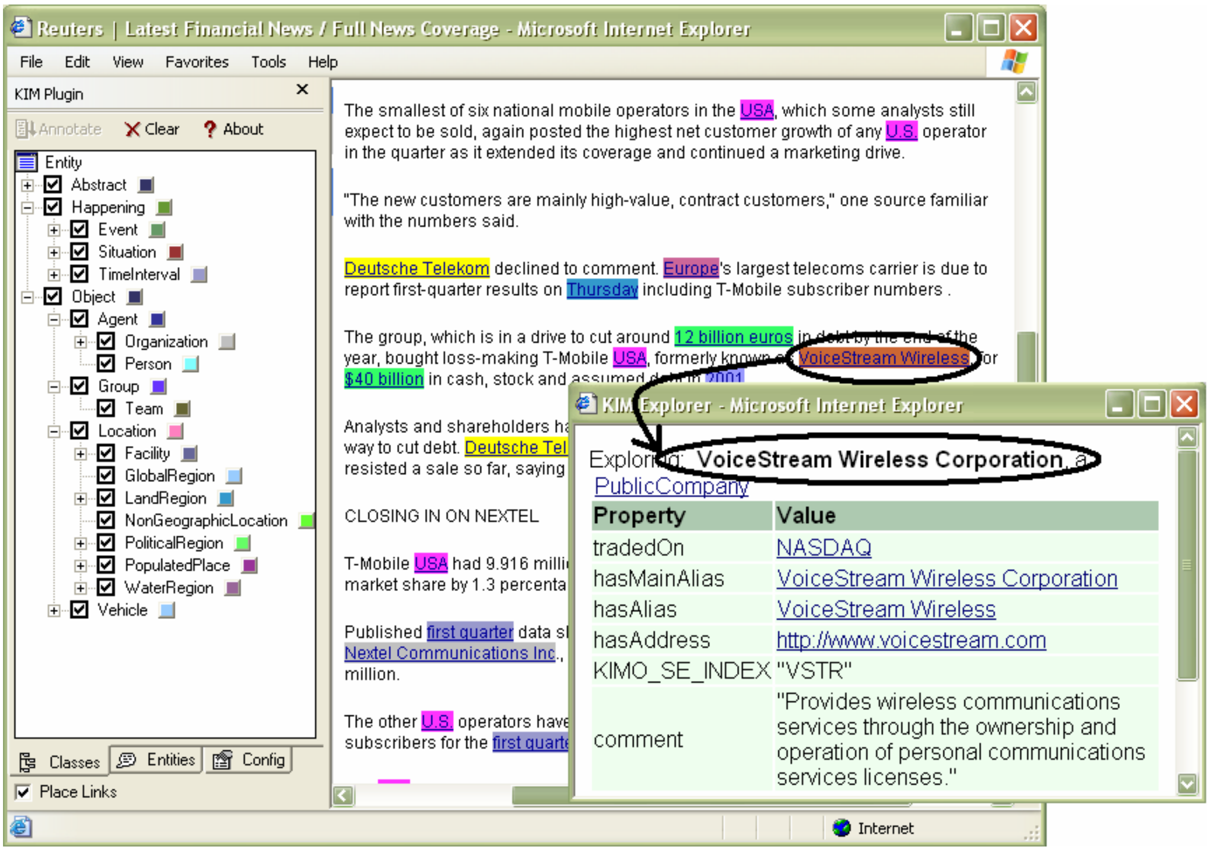

XYZ announced profits in Q3, planning to build a $120M plant in BULGARIA, and more and more and more and more text.
Semantic Repository
Company
Location
City
Country
type
type
type
XYZ
locOn
London
Bulgaria
establOn
partOf
type
"03/11/1978"
UK

Figure 2. Semantic annotation

The semantic annotation process can be seen as a classical named entity recognition and annotation process. The named entity type is specified by reference to an ontology, and the semantic annotation requires identification of the entity. The approach of semantic annotation comprises two processes: (a) information extraction and (b) identity resolution.

[1] *Named-entity* stands for a phrase that clearly identifies one item from a set of other items that have similar atributes (person names, company names, geographic locations, etc.)


Reuters | Latest Financial News / Full News Coverage - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
KIM Plugin
Annotate
Clear
About
Classes
Entities
Config
Internet
KIM Explorer - Microsoft Internet Explorer

Entity
☑ Abstract
☑ Happening
☑ Event
☑ Situation
☑ TimeInterval
☑ Object
☑ Agent
☑ Organization
☑ Person
☑ Group
☑ Team
☑ Location
☑ Facility
☑ GlobalRegion
☑ LandRegion
☑ NonGeographicLocation
☑ PoliticalRegion
☑ PopulatedPlace
☑ WaterRegion
☑ Vehicle

☑ Place Links

The smallest of six national mobile operators in the USA, which some analysts still expect to be sold, again posted the highest net customer growth of any U.S. operator in the quarter as it extended its coverage and continued a marketing drive.

"The new customers are mainly high-value, contract customers," one source familiar with the numbers said.

Deutsche Telekom declined to comment. Europe's largest telecoms carrier is due to report first-quarter results on Thursday including T-Mobile subscriber numbers .

The group, which is in a drive to cut around 12 billion euros in debt by the end of the year, bought loss-making T-Mobile USA, formerly known as VoiceStream Wireless, for $40 billion in cash, stock and assumed debt in 2001.

Analysts and shareholders ha
way to cut debt. Deutsche Tel
resisted a sale so far, saying

CLOSING IN ON NEXTEL

T-Mobile USA had 9.916 milli
market share by 1.3 percenta

Published first quarter data s
Nextel Communications Inc.,
million.

The other U.S. operators have
subscribers for the first quarte

Exploring: VoiceStream Wireless Corporation, a PublicCompany

Property	Value
tradedOn	NASDAQ
hasMainAlias	VoiceStream Wireless Corporation
hasAlias	VoiceStream Wireless
hasAddress	http://www.voicestream.com
KIMO_SE_INDEX	"VSTR"
comment	"Provides wireless communications services through the ownership and operation of personal communications services licenses."

Figure 3. Semantic annotation in use

2.2 Information Extraction

Information extraction refers to the lightweight process of detecting pieces of relevant informational units in texts and representing them in the form of attribute/value templates [3][4]. For example, the information extraction engine will detect the expression “event in London” and place the word “London” in the respective template as the value for the attribute “location”.

Some natural language processing techniques explore particular parts of the texts to identify *named entities* such as:

- organization names,
- person names,
- geographical locations,
- monetary units,
- time expressions

Named Entity Recognition (NER) is a subtask of the information extraction process, which identifies and classifies language expressions into predefined categories, and annotates the input text with the recognized categories.

There are different approaches to information extraction:

- *Statistical approaches*, where Machine Learning techniques are employed in order to train a statistical model for NER. The main advantage of this approach is that the NER model can be bootstrapped relatively quickly, even with a limited training corpus. The main disadvantage of this approach is that the quality of the results depends on the data in the training corpus, e.g. if some patterns are not present in the training corpus, then naturally the final NER model will not be able to recognize them in new texts.

- *Rule-based approaches*, where some sort of a rule base is created in order to describe the most common patterns that the NER system should recognize. The patterns usually are based on regular expressions, though the specific pattern language may provide many additional extensions and capabilities. The main advantage of this approach is that the rule base compilation process does not need training corpora, e.g. the respective expert may describe rules even for patterns that are not available in a specific training corpus. The main disadvantage is that the rule compilation process is much slower and more expensive than a machine learning based model training.

- *Hybrid approaches*, a combination of the above approaches, where usually a statistical model is built first from a training corpus, and then rules are automatically generated for the language expert to review and extend. This is the optimal cost/benefit approach for NER systems.

The information extraction process identifies the text chunk, the knowledge element and proceeds to identity resolution by matching the result to the instance information for each known named entity in the text. It adds the new entities with their semantic descriptions and relations to the knowledge base. Thus, as a result each named entity is linked to its type and to its individual semantic description.

Other natural language processing techniques work on the entire text and analyze all words and relationships between them in order to identify facts, events and happenings along with the named entities. They all process all words in texts, build structures and interpret their meaning by giving rise to different actual applications. They differ from the previously mentioned approach of key-word spotting and named entity recognition, because their main purpose is to understand the meaning of the text and capture as many concepts and relations between words as possible. Such natural language processing techniques produce systems that are able to interpret full natural language expressions and can be seen as complimentary to the semantic web, which categorizes and classifies content and brings it to unique identifiers. Such systems produce an interpretation layer between raw text and data represented in ontologies of the sort of DBpedia and all Linking Open Data by handling the full stream of natural language input to identify metadata.

2.3 Identity Resolution

The process of *identity resolution* (also called *entity reconciliation*) has the goal to determine whether two or more data representations refer to the same object and thus should be resolved into one representation. Identity resolution looks at addresses, names, social security numbers, dates, and customer history to make interconnections between different identity records. The process of identity resolution is closely related to the ontology development and use. The named entity references in the text are linked to an entity individual in the knowledge base. The entity instances all bear unique identifiers that allow the annotations to be linked to the exact individual in the knowledge base and thus each recognized named entity is linked to an individual in the KB and the associated semantic description.

Identity resolution is necessary in the process of data acquisition from multiple data sources or knowledge bases in order to ensure that all the information related to an entity instance is aggregated together and there are no duplicate objects in the knowledge base referring to the same instance.

2.4 Sentiment mining

Sentiment analysis or opinion mining aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be his or her judgement or evaluation, affective state (the emotional state of the author when writing), or the intended emotional communication (the emotional effect the author wishes to have on the reader). A basic task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level — whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral.

Automated sentiment analysis of digital texts utilizes elements from machine learning such as latent semantic analysis, support vector machines, "bag of words"[2]. More sophisticated methods try to detect the holder of a sentiment (i.e. the person who maintains that affective state) and the target (i.e. the named entity or target whose affective state one is interested in). To mine the opinion in context and get the feature which has been opinionated, the grammatical relationships of words are used. Grammatical dependency relations are obtained by deep parsing of the text [2].

Open source software tools deploy machine learning, statistics, and natural language processing techniques to automate sentiment analysis on large collections of texts, including web pages, online news, internet discussion groups, online reviews, web blogs, and social media.

The rise of social media such as blogs and social networks has fueled interest in sentiment analysis. With the proliferation of reviews, ratings, recommendations and other forms of online expression, online opinion has turned into a kind of virtual currency for businesses looking to market their products, identify new opportunities and manage their reputations. As businesses look to automate the process of filtering out the noise, understanding the conversations, identifying the relevant content and actioning it appropriately, many are now looking to the field of sentiment analysis.

[2] http://en.wikipedia.org/wiki/Bag_of_words_model_in_computer_vision

3 Text Mining Platforms

This section provides a brief overview of two platforms for text mining and semantic annotation – GATE and KIM. The goal of the section is not to provide a comparison of the various opens source or commercial systems, but to outline the features of the two selected platforms which make them suitable for the basis of ETL from unstructured data sources of the CUBIST use cases.

3.1 GATE (General Architecture for Text Engineering)

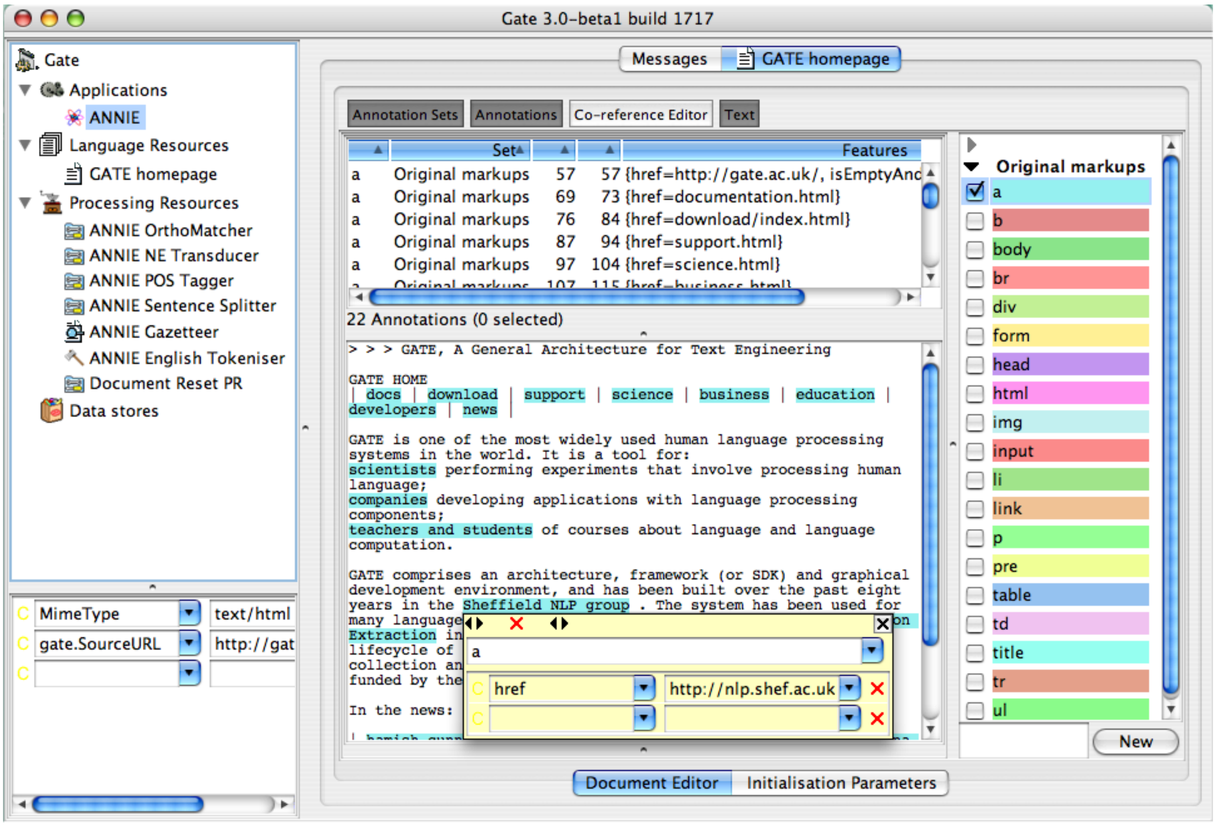
GATE[3] is a platform for developing and deploying text mining software components. GATE is open source software (distributed under a LGPL license[4]); users can obtain free support from the user and developer community or on a commercial basis from the ecosystem of industrial partners.

GATE comes in several editions:

- *GATE Developer*, an Integrated Development Environment (IDE) for language processing components bundled with a very widely used Information Extraction system and a comprehensive set of 3rd party plugins (Figure 4)
- *GATE Teamware* (web application), a collaborative annotation environment for factory-style semantic annotation projects built around a workflow engine and a heavily optimised backend service infrastructure
- *GATE Embedded* (framework), an object library optimised for inclusion in diverse applications giving access to all the services used by GATE Developer

[3] http://gate.ac.uk/

[4] http://www.gnu.org/licenses/lgpl.html



MimeType	text/html
gate.SourceURL	http://gat

	Set			Features
a	Original markups	57	57	{href=http://gate.ac.uk/, isEmptyAnc
a	Original markups	69	73	{href=documentation.html}
a	Original markups	76	84	{href=download/index.html}
a	Original markups	87	94	{href=support.html}
a	Original markups	97	104	{href=science.html}
a	Original markups	107	115	{href=business.html}

22 Annotations (0 selected)

```
> > > GATE, A General Architecture for Text Engineering

GATE HOME
| docs | download | support | science | business | education |
developers | news |

GATE is one of the most widely used human language processing
systems in the world. It is a tool for:
scientists performing experiments that involve processing human
language;
companies developing applications with language processing
components;
teachers and students of courses about language and language
computation.

GATE comprises an architecture, framework (or SDK) and graphical
development environment, and has been built over the past eight
years in the Sheffield NLP group . The system has been used for
many language
Extraction in
lifecycle of
collection an
funded by the

In the news:
```

a

href	http://nlp.shef.ac.uk

Original markups: ☑ a, □ b, □ body, □ br, □ div, □ form, □ head, □ html, □ img, □ input, □ li, □ link, □ p, □ pre, □ table, □ td, □ title, □ tr, □ ul

Figure 4. GATE Developer

On top of the core functions GATE includes components for diverse language processing tasks, e.g. parsers, morphology analysers, part-of-speech taggers, information retrieval tools, information extraction components for various languages, etc. GATE Developer and GATE Embedded are supplied with a standard Information Extraction pipeline (ANNIE).

3.1.1 CREOLE: the GATE Component Model

The GATE architecture is based on components: reusable chunks of software with well-defined interfaces that may be deployed in a variety of contexts. The design of GATE is based on an analysis of previous work on infrastructure for Language Engineering. GATE components are one of three types:

- *LanguageResources* (LR) representing entities such as lexicons, corpora or ontologies;
- *ProcessingResources* (PR) representing entities that are primarily algorithmic, such as parsers, generators or n-gram modellers;
- *VisualResources* (VR) representing visualisation and editing components that participate in GUIs.

The GATE framework performs these functions across all components and resources:

- component discovery, bootstrapping, loading and reloading;
- management and visualisation of native data structures for common information types;

- generalised data storage and process execution.

GATE comes with various built-in components on board:

- Language Resources modelling Documents and Corpora, and various types of Annotation Schema
- Processing Resources that are part of the ANNIE system
- Gazetteers
- Ontologies
- Machine Learning resources
- Alignment tools
- Parsers and taggers

3.1.1.1 Tokenizers

The tokenizer component splits the text into very simple tokens such as numbers, punctuation and words of different types. For example, it distinguishes between words in uppercase and lowercase, and between certain types of punctuation.

3.1.1.2 Gazetteers

The role of the gazetteer is to identify entity names in the text based on predefined lists. The gazetteer lists may be generated either from plain text files or directly from the knowledge base. Each list represents a set of names, such as names of cities, organisations, locations, days of the week, etc.

3.1.1.3 Sentence Splitter

The sentence splitter is a cascade of finite-state transducers which segments the text into sentences. This module provides input for the Part-of-speech (POS) Tagger. The splitter uses a gazetteer list of abbreviations to help distinguish sentence-marking full stops from other kinds.

3.1.1.4 POS Tagger

The POS tagger produces a part-of-speech tag (such as “V” for verbs, “ADJ” for adjectives or “NP” for proper nouns) as an annotation on each word or symbol.

3.1.1.5 OrthoMatcher

The OrthoMatcher module adds identity relations between named entities found by the semantic tagger, in order to perform coreference resolution within the document. It does *not* find new named entities as such, but it may assign a type to an unclassified proper name if it was matched to an already classified name within the text.

3.1.2 JAPE component

JAPE is the Java Annotation Patterns Engine part of GATE. It provides finite state transduction over annotations based on regular expressions. JAPE makes it possible to recognise complex regular expressions in annotations on documents.

A JAPE grammar consists of a set of phases, each of which consists of a set of pattern/action rules. The phases run sequentially and constitute a cascade of finite state transducers over annotations. The left hand side (LHS) of the rule contains the identified annotation pattern that may contain regular expression operators (e.g. *, ?, +). The right hand side (RHS) outlines the action to be taken on the detected pattern and consists of annotation manipulation statements. The LHS of a JAPE grammar aims to match the text span to be annotated, whilst avoiding undesirable matches.

Example: detecting team names, based on name of the city followed by certain suffixes, i.e. “City + United”, “City + F.C.” or “City + FC”

```
Rule: team_rule_01
Priority:50
(
     {City}
     (
        {Token.string=="United" } | {Token.string=="F.C." } | {Token.string=="FC" }
     )
):team

-->

:team.Team = {rule= " team_rule_01" }
```

3.2 KIM

KIM[5] is a software platform for automated semantic annotation, indexing, and retrieval of unstructured and semi-structured content. The most popular use cases for KIM are:

- Generation of meta-data for the Semantic Web, which allows hyper-linking and advanced visualization and navigation.
- Semantic search over unstructured and semi-structured content.

KIM analyzes textual content and identifies references to entities (persons, organizations, locations, dates, etc.) or the relations that exist between entities (such as job positions). Then it tries to match the discovered entities the already known entities in the knowledge base. Finally, the original documents are enriched with metadata about the entities and relations that they contain. The whole process is referred to as *semantic annotation* (see Figure 2 and Figure 5).

[5] http://www.ontotext.com/KIM

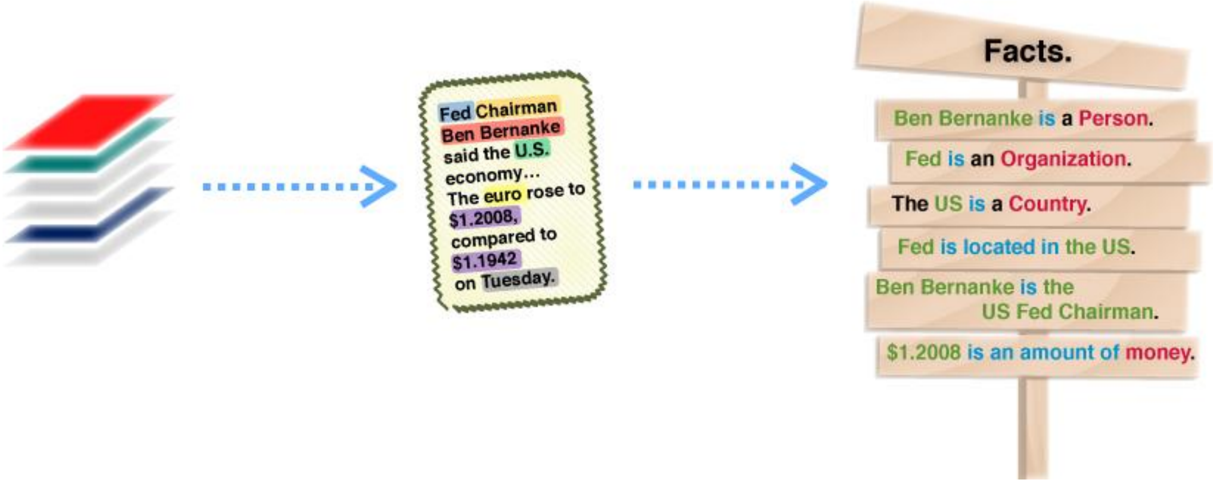


Figure 5. Semantic information extraction

In order to allow the easy bootstrapping of semantic annotation applications, KIM is equipped with a small upper-level ontology called PROTON[6]. It is comprised of about 250 classes and 100 properties. Furthermore, KIM provides a knowledge base pre-populated with about 200,000 entity descriptions for most popular entities (countries, cities, politicians, organizations, famous people, etc). Its role is to provide background knowledge that improves the semantic annotation process.

KIM incorporates several popular open-source platforms:

1. GATE (described in Section 3.1)

2. Sesame[7]: An RDF repository and framework. It provides the Storage and Inference Layer (SAIL) which allows RDF databases (including OWLIM) to be plugged on top of Sesame. Most commercial RDF databases at present provide a SAIL compatibility interface.

3. Lucene[8]: An open-source Information Retrieval engine

KIM is a highly adaptable and modular platform for linking and navigating data, content, and knowledge. It can be configured to use all or some of its components to suit different needs. Computationally intense components such as concepts extraction and semantic database can be clustered to reach the performance you need.

Figure 6 shows a typical architecture of a KIM-based system.

[6] http://proton.semanticweb.org/

[7] http://www.aduna-software.com/technology/sesame

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

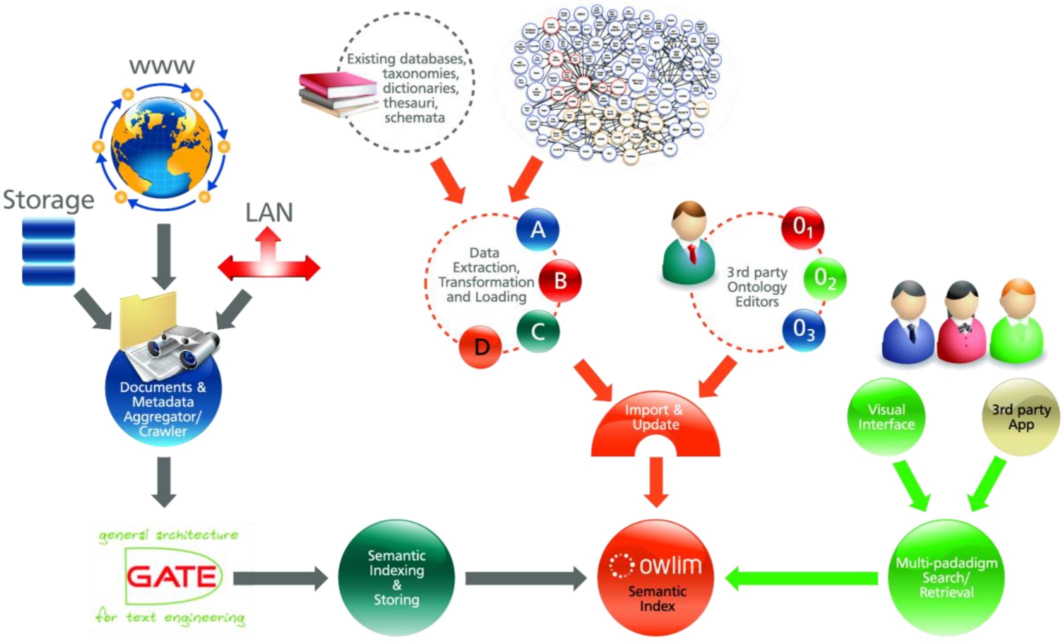

www
Storage
LAN
Documents & Metadata Aggregator/ Crawler
Existing databases, taxonomies, dictionaries, thesauri, schemata
Data Extraction, Transformation and Loading
A
B
C
D
3rd party Ontology Editors
O1
O2
O3
Import & Update
Visual Interface
3rd party App
general architecture
GATE
for text engineering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

Figure 6. Semantic annotation and search with KIM

The workflow is usually comprised of the following steps:

1. *ETL from structured data sources* – existing structured data sources are RDF-ized and data is stored in the semantic repository (RDF database). If the structured data sources are already in RDF (for example Linked Open Data sets) then the ETL step is simplified, since only ontology/vocabulary alignment and mapping is required.

2. *Semantic annotation of text content* – existing unstructured data sources are analysed (using the information extraction and text mining techniques outlined in Section 2) so that the entities and the relationships between the entities found in text can be identified. This step enriches the unstructured data sources with semantic annotations

3. *Semantic indexing & interlinking* – the semantic annotations generated in step 2 are stored in the semantic repository (RDF database). Since new data (in forms of explicit RDF triples) is added to the database, the built-in reasoner in the database may infer even more (implicit) triples and add them to the database as well. The end result of this phase is that all entities discovered in step 2 as well as the links (relations) between entities or between entities are stored in the database as RDF triples.

4. *Semantic search* – the result of steps 1-3 is a uniform representation of structured and unstructured content and the entities and relations between entities discovered in the content. This allows for more complex, multi-paradigm semantic search over the content, which goes beyond traditional structured search (SQL) or keyword / full-text search.

4 ETL from unstructured data sources in CUBIST

Based on the analysis from Sections 2 and 3 this section will provide an outline for the work in Task 2.1 of CUBIST, with respect to the specific requirements of the CUBIST use cases.

4.1 Use case requirements

CUBIST D3.1.1 already provides a brief description of the various data sources that the three CUBIST use cases provide. A brief summary of the outcome of D3.1.1 follows:

- The HWU use case does *not* contain any unstructured data, since all the EMAP/EMAGE data is stored in (structured) databases. There is an option of analysing textual content from scientific journal publications related to EMAGE, but since most of this information is not publicly accessible, this is only an optional requirement for the HWU use case.
- The SAS use case provides a mixture of structured and unstructured data sources (with predominantly *structured* ones). The unstructured data sources are usually in the form of human generated notes, messages or communication transcripts. The SAS systems currently in use do not analyse the unstructured content. The requirement for the SAS use case within CUBIST is to identify entities and relations in the text content and to interlink it with the data from the structured data sources in order to provide more expressive search capabilities. Valuable features of the Semantic ETL system would be support for co-reference resolution and fuzzy matching.
- The INN use case also provides a mixture of structured and unstructured data sources (with predominantly *unstructured* ones). The unstructured data consists of vacancy description, crawled company web pages, public domain news sources, company forums, social streams such as twitter. This is also the use case which requires the most advanced types of text mining, going beyond the traditional entity and relation extraction, e.g. automated job categorisation, sentiment and opinion analysis from social streams, automated skill and qualification extraction.

4.2 Applicability of existing platforms

Section 3 provides a summary of the features of the GATE and KIM platforms related to text mining and semantic annotation of unstructured data. The combination of the two platforms provides a solid basis for (Figure 6):

- Semantic annotation of text content
- Semantic indexing and interlinking of entities and relations extracted from text
- Multi-paradigm semantic search

The GATE and KIM platforms have already been applied in use cases dealing with various types of unstructured data sources: emails, documents in various formats, wikis, web pages, Twitter streams, etc. This fits very well in the requirements of the SAS use case, where the focus is mostly on traditional text mining and information extraction over unstructured data. For the INN use case, in addition to the semantic annotation provided by GATE and KIM, several new text mining related features will be developed within CUBIST.

4.3 New features to be developed within CUBIST

In addition to adapting the GATE and KIM text mining and semantic annotation platforms to the two CUBIST use cases which deal with unstructured information (INN and SAS), the integrated prototype D2.3.1 (Data Integration and Federation Platform) will provide functionality supporting the newly identified requirements of the INN use case, regarding the qualification extraction and sentiment mining.

4.3.1 Qualification extraction

Qualification extraction from online resumes, as defined in the requirements of the INN use case, can be considered as a multi-label classification task, where each input document (resume) is classified according to one or more categories (classes in a predefined qualification taxonomy). The major focus of this task will be in training a statistical model according to the particular input data of the INN use case.

For the actual statistical model we will use a character-level N-gram approach [5] that avoids the difficulties associated with the handling of domain- and language- specific feature extraction, thus providing a very good balance between accuracy and scalability of the implementation.

In addition to the baseline approach described in [5], the qualification extraction module developed within D2.3.1 will incorporate useful metadata from the provided documents, as well as the named entities extracted from text in order to increase the accuracy of the classification process. Additionally, various smoothing techniques can be applied to improve the performance of the statistical model.

4.3.2 Sentiment mining

Section 2.4 outlined the major goals related to mining of sentiments and opinions in text. In accordance with the provided requirements, our initial goal will be the implementation of a system for polarity mining that would be able to detect the positive or negative sentiment expressed toward the activities of an organization. Similarly to the qualification extraction module, the sentiment mining component to be developed within D2.3.1 will be based on a statistical model, however one that utilizes a number of advanced natural language processing techniques that enable meaning analysis. Presently, we are capable of implementing a large part of the methods described in [6], and we can choose the most appropriate approach upon receiving a more specific set of constraints for the task. In addition to the basic textual features used in the majority of the described methods, the sentiment mining statistical model can further be improved by incorporating named entities.

5 Conclusion

This deliverable provides the first version of the analysis of CUBIST use case requirements for extracting information from unstructured data sources as well as existing text platforms which can support these requirements.

Work on extracting information from unstructured data sources within CUBIST will continue with the first integrated prototype of the CUBIST data integration and federation platform (D2.3.1) as well as the second, extended version of this analysis (D2.1.2)

The first concrete results will be provided within D2.3.1 by the M18 milestone, where the ongoing effort is focused in two main directions:

- Adapting the GATE and KIM text mining and semantic annotation platforms to the particular CUBIST use cases
- Extending the platforms with functionality for qualification extraction and sentiment mining (required by the INN use case)

6 References

[1] Peter Turney (2002). "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews". *Proceedings of the Association for Computational Linguistics*. pp. 417–424

[2] Lipika Dey , S K Mirajul Haque (2008). "Opinion Mining from Noisy Text Data". *Proceedings of the second workshop on Analytics for noisy unstructured text data, p.83-90.*

[3] Sarawagi, S. “Information Extraction”, *Foundations and Trends® in Databases*, 2008, *1*, 261-377

[4] Grishman, R. “Information Extraction: Techniques and Challenges”, *International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, Springer-Verlag*, 1997, 10-27

[5] Peng, F., Shuurmans, D., Wang, S. (2003) “Language and Task Independent Text Categorization with Simple Language Models”. *Proceedings of HLT-NAACL’2003, p.110-117*

[6] Yelena Mejova (2009) “*Sentiment Analysis: An Overview*” http://www.divms.uiowa.edu/~ymejova/publications/CompsYelenaMejova.pdf