Grant Agreement Number: 257528

KHRESMOI

www.khresmoi.eu

Report on reference identification component

Deliverable number	*D1.5*
Dissemination level	*Public*
Delivery date	*31 August 2013*
Status	*Final*
Author(s)	*Konstantin Pentchev, Andrey Avramov, Vassil Momtchev, Angus Roberts, Johann Petrak*

This project is supported by the European Commission under the Information and Communication Technologies (ICT) Theme of the 7th Framework Programme for Research and Technological Development.

Abstract

This deliverable presents an approach for linking named entities from information extraction to resources in the Khresmoi Knowledge Base. A novel approach is developed for disambiguating terms in the text and their semantic interpretation based on a user-defined logical model of the source documents. The methodology was evaluated on a corpus of web pages from a crawled HON certified web site. The comparison to a manually annotated subset of the pages confirmed that the method achieves highly precise results with a small trade-off for lower recall.

Table of Contents

List of Abbreviations

CT	Computed Tomography
GAPP	GATE Application Processing Pipeline
IE	Information Extraction
KB	Khresmoi Knowledge Base
KS	Large Scale Biomedical Knowledge Server
NEC	Not Elsewhere Classified
NER	Named Entity Recognition
NOS	Not Otherwise Specified
PET	Positron Emission Tomography
PR	(GATE) Processing Resource
RDF	Resource Description Framework
SBT	Semantic Biomedical Tagger
SKOS	Simple Knowledge Organization System
TOS	Talend Open Studio
UI	User Interface
UMLS	Unified Medical Language System
URI	Universal Resource Identifier

1 Executive summary

This deliverable presents our proposition for a generic solution that links GATE annotated documents to the Khresmoi Knowledge Base (KB). The work is based on the approach and results that were presented in D1.4.1 "Report accompanying manually annotated reference corpus" [1] and integrates the KB infrastructure and data. Specifically, it proposes and evaluates a methodology for performing semantic named entity recognition with a novel disambiguation approach tailored for the bio-medical domain.

D1.4.1 addresses the creation of a manually annotated corpus, where annotations or tags are assigned to selected parts of the text. The meta-data includes relevance for the sections and the type of the contained entities. Ultimately, we would like to map the generated meta-data to a semantic model and link it to the existing structured knowledge. However, there may be several entities in the knowledge base that standard approaches find suitable for referencing given one and the same span of text. We expanded on the work in D1.4.2 [2] to develop a technique for tackling this issue. The approach is based on modelling the logical structure of the different document types that are to be annotated and doing differential semantic named entity recognition based on these meta-models.

This deliverable describes:

- the approach used to disambiguate the assigned annotations and link them to knowledge base identifiers (URIs).
- the methodology used to segment documents into different sections and better contextualise the meaning of terms (for example, "headache" in the symptoms of a disease section is a symptom whereas "headache" in the possible side effects section is an adverse drug event).
- the software infrastructure that controls the linkage of the Gate document annotations to the semantic model, based on document modelling.

In order to test the performance of the methodology, it was applied to a selected corpus of web pages relevant to the Khresmoi project. The results of the reference identification were evaluated against a manually annotated test set of the pages. It achieved a precision of 97% and a recall of 62%. We consider these results to be a success, as our aim, with regard to the project goals, was to extract knowledge with high trustability. Hence, we plan to apply the method to a wider set of documents and use the reference to the KB to enhance the search performance of the Khresmoi system.

2 Introduction

The goal of the Khresmoi project is to provide a multi-lingual search engine for biomedical content [3]. As described in D1.4.2 [2], in order to implement non-trivial search functionality over the text resources semantic information extraction [4] techniques are applied. This involves linking of terms recognized in texts to the resources in the Khresmoi KB [5][6]. However, one span of text is often marked with several annotations. For example, in biomedical texts a protein and a gene can share the same name, resulting in multiple overlapping annotations. In order to assign the correct meaning to a term, disambiguation techniques need to be applied. In the context of semantic annotations, the aim is to assign each extracted entity a single correct reference to an entity from structured knowledge.

One approach to resolve this ambiguity is to use contextual information. Methods using machine learning in order to obtain term co-occurrence statistics [7] and type contextual evidence [8] have been widely applied with varying precision (in some cases reaching the 90% margin). However, these methods depend on either an extensive manually annotated gold standard corpus or on the presence of domain specific class labels in the context itself.

An alternative is to use knowledge-intensive approaches, based on domain specific lexicons and semantic models. These techniques have been previously applied to NER in open-domain texts by using the semantic relations from publicly available sources such as Wikipedia [9][10]. These approaches perform well when disambiguating between entities from different domains that share labels. However, when the entities are from the same domain, in this case the biomedical one, choosing a correct instance has to be based on more specific discrimination. Consider a clinical study report that discusses the measurements of patient's haemoglobin. There are at least two interpretations of the subject – as the protein molecule and as a biomarker. While the former annotation is not strictly incorrect, it may prevent us from extracting valuable knowledge if our goal is to gather all the relevant study measurements performed.

In Section 4 we describe a different approach, which makes use of pre-defined semantic sectioning, i.e. extracting the logical structure of a document and mapping it to a meta-model, and disambiguation at the gazetteer creation stage in order to achieve correct interpretation of the named entities. In Section 5 we present the implemented infrastructure, which allows domain experts to define document templates and perform semantic annotations according to our methodology. Finally, in section 6 we present the results of processing a selected corpus of HON certified pages[1] (described in Section 3) and evaluate the results against manual annotations.

[1] http://www.hon.ch

3 Corpus

This section describes the set of documents we selected for the evaluation of the methodology proposed in section 4. We describe in detail the document inclusion criteria, document structure and content statistics.

3.1 Inclusion criteria

In order to demonstrate our methodology, we aimed to select a corpus from HON certified web sites. HON provided us with a service for accessing their crawled pages, which have some meta-data attached to them. We did not sample the pool randomly, as our methodology is aimed at giving precise IE results, which is based on user input about the documents. While this makes the method less generic, experience has proven that data quality is of the highest importance for the industry. In this deliverable we present the study of how the approach performs. However, we plan to expand the domains processed for future Khresmoi prototypes. Therefore, we have applied the following criteria when selecting the documents:

- The documents should have a clearly identifiable structure
- The sections naming should be consistent, which will allow accurate mapping to their relevant semantic sections.
- Some of the sections should contain information, which can be looked up by our gazetteers, i.e. biomedical data, preferably about diseases, drugs, symptoms, etc.

3.2 Corpus description

Based on these criteria a set of 2343 drug notes were retrieved from allinahealth.org. Most of the documents contained eight different types of sections. Each section type had exactly the same naming and it was placed within a <H3> tag that made section boundaries definition extremely clean. In addition, two unnamed sections were discovered – one within the first <H2> tag that covered the drug name and another within the second <H4> tag that covered the indication of that drug. Six types of sections looked interesting in the scope of named entity recognition with gazetteers.

4 Methodology

This section describes in detail the methodology behind our approach. Our goal was to implement a generic process, which - based on the semantic sectioning of documents - could perform automatic document classification. The successfully classified documents were used as input to create semantic annotations that were subsequently linked to resources in the semantic repository (i.e. the KB). Both the sectioning and annotating steps were developed as GATE application processing pipelines [11] and each of them was built up using a specific collection of GATE processing resources [11]. As the methodology depends on the user input for the document meta-model, in Section 4.5 we describe a methodology for general disambiguation of UMLS terms, which can be used with any document, relevant to the Khresmoi use-cases.

4.1 Document meta-model

We define *syntactic sectioning* as the segmentation of a textual document into a tree of distinct parts, based on the structural and syntactic features of the latter – e.g. accented styles, font size, specific phrases in section title, etc. The resulting structure is a tree because the different structural parts exhibit a nesting pattern – section A can have a sub-section B, which in turn can contain many sub-sub-sections.

Semantic sectioning is the process of mapping distinct parts of text, usually identified through syntactic sectioning, to a set of pre-defined categories that represent the document's logical structure. However, there is no universal document structure, even for documents from the same domain, with similar goals, etc. Therefore, the specific semantic sections for a document type have to be explicitly defined prior to performing information extraction. We call this formal description of a document's logical structure the document meta-model. The meta-model allows us not only to execute specific annotation pipelines over specific parts of the document, but also to do more precise semantics extraction, e.g. if a disease resource X is found in the indication section of document A and in the contra-indication section of document B, it is obvious that it has very different semantics in the context of these documents.

Because we aimed at performing high precision semantic annotation, it was important to devise a methodology that allows us to specifically map syntactic sections to semantic sections, while at the same time allowing flexibility to define different rules for performing the segmentation over different classes of documents. In order to achieve this, a generic segmentation processing resource was developed that uses regular expressions to identify sections. However, the actual regular expressions are not defined in the PR but are specified in the meta-model. The meta-model is then loaded as an initialisation parameter of the pipeline. In addition, you can have not only one document class but many, which re-use certain PRs. Therefore, the shareable PRs – a set of gazetteers, each populated with a different vocabulary (hence referred to as extraction types) – are also described in the meta-model. A formal description of the meta-model is given in Figure 1.

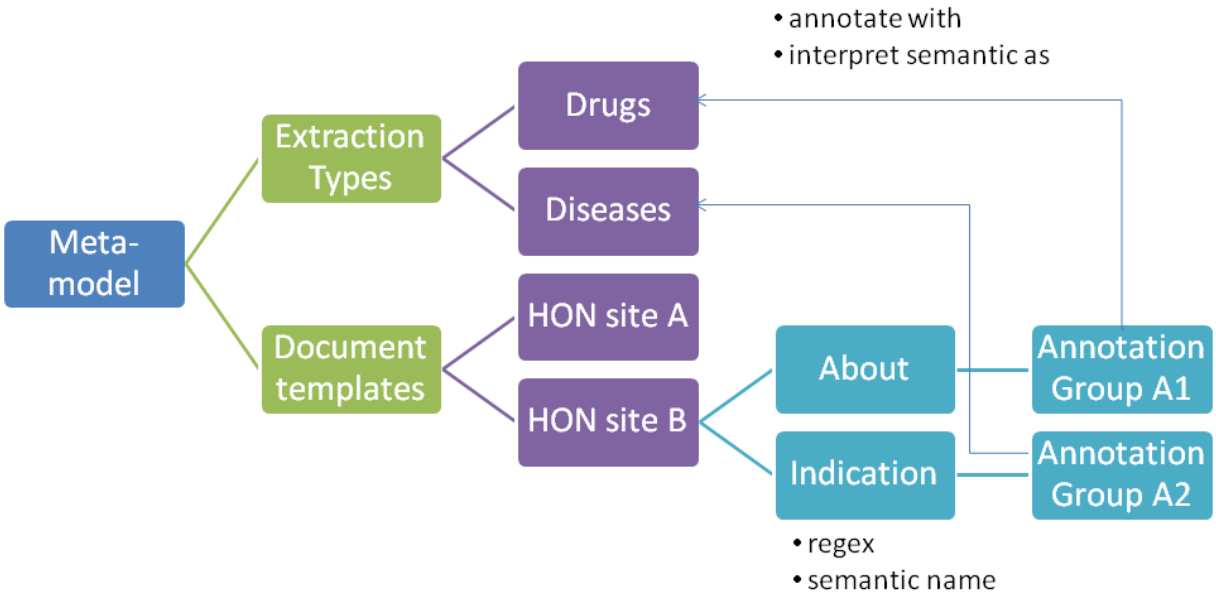

Figure 1 : Formal presentation of the document meta-model. It is conceptually divided in predefined extraction types (gazetteers) and specified document templates (classes). Each template contains a tree of semantic section definitions ("About", "Indication"), which are mapped to syntactic sections by a regular expression definition. Each section is also linked to a set of extraction types through the so-called annotation groups, which specify also the semantic interpretation.

4.2 Classification

Classification is the process of assigning a document to a pre-defined document class from the meta-model. This is done based on the semantic sections from each template identified in the text.

The classification process is implemented as part of the sectioning GAPP. It is initialised with the following parameters:

- encoding = "UTF-8"
- markupAware = true
- mimeType = "application/xhtml+xml"
- preserveOriginalContent = false

There are two types of annotations set by the GAPP: document features and section annotations. Currently, some of the document features are dynamic, i.e. they are produced according to the meta-model. All these features are of TYPE_XXX_SCORE type, where XXX corresponds to Template Names defined in the system. Each TYPE_XXX_SCORE decimal value represents the number of matching regular expressions from a template against the input document. The formula used for calculating the score is 2*M/(A+C), where M is the number of the mapped sections in the input document to a given document template; A is the number of all syntactic sections, found in the document; C is the number of all semantic sections for a given document template. The highest TYPE_XXX_SCORE for each document is used to determine the document feature TYPE.

We prepared one document template for the evaluation named AllinaHealth DrugNotes with 6 semantic sections. Details on the section meta-model are given in Table 1.

Semantic section name	Regex for segmenting
Proprietary Drug Name	* [^<]+ *
Indication	* [^<]+ *
Other Drug Names	Brand Name(s)
Drug Interactions	Drugs and Foods to Avoid
Adverse Events	Possible Side Effects While Using This Medicine
Contradictions	When This Medicine Should Not Be Used

Table 1 : The semantic sections for the *AllinaHealth DrugNotes* document template with corresponding regular expressions. The six sections defined are all of the same level (siblings).

4.3 Named Entity Recognition

Once the document class and semantic sections are determined, we set up an annotation GAPP to extract appropriate information. In Table 2 we provide a detailed description of the processing components in the GAPP.

Processing Resource	Description
Tokeniser	Standard GATE tokeniser
Sentence Splitter	GATE regular expression based sentence splitter
POS Tagger	GATE POS tagger trained on biomedical corpus (GENIA)
Morphological Analyser	FLEX based morphological analyser
Segmented Annotations	Governs which Gazetteer should be run on which semantic sections
Drugs Gazetteer	Gazetteer populated from DrugBank
Diseases Gazetteer	Populated with UMLS concepts from 14 semantic types related to diseases and body parts (T019, T020, T022, T023, T029, T030, T046, T047, T048, T049, T050, T184, T190, T191)

Table 2: Processing resources used by annotation GAPPs. The distinct Gazetteers are always part of separate GAPPs.

In order to extend the gazetteers' abilities to match not only the original but also derivative text chunks that did not exist in the KB, we applied a set of rewrite rules that are applied to each label entering the gazetteer during its population (similar to the rules described in [2]) :

- Roots of the words were determined.
- A set of rewrite rules were applied:
    - the labels were filtered as follows:
        - filter out labels that contain an at (@) sign
        - filter labels that contain “not otherwise specified”, “unspecified” “[NOS]” and similar
        - filter labels that contain “NEC”, “not elsewhere classified”, “unclassified” and similar
        - filter very short labels
    - derivative labels were created in the following ways:
        - remove angular brackets
        - remove multiple spaces
        - remove possessives
        - remove brackets at the end
        - remove parentheses at the end
        - invert labels that have a single comma: e.g. “pain, dorsal” → “dorsal pain”
    - labels with 6 or more tokens were removed.

The text chunks that had to be compared to the gazetteers’ content were “rooted” as well.

Due to the specificity of the biomedical knowledge domain and the knowledge source used (UMLS), many of the literals that had to be stored in the Diseases gazetteer were related to more than one concept. This problem could be at least partially fixed by disambiguation priority mechanism elaboration. The implemented disambiguation mechanism was based on two assumptions:

- Each instance has one preferred label and all preferred labels are unique in the UMLS Metathesaurus.
- Each instance has zero or more alternative labels and each of them has one or more sources.

Since the ambiguity was caused by the duplication of alternative labels for different instances and the simultaneous string and root usage as alternative token features, gazetteers population was done by applying an eight-stage priority mechanism summarised in Table 3:

1. Strings matching preferred labels were annotated.
2. Strings matching alternative labels with the highest number of distinct sources were annotated.
3. Roots matching preferred labels were annotated.
4. Roots matching alternative labels with the highest number of distinct sources were annotated.
5. Strings matching rewritten preferred labels were annotated.
6. Strings matching rewritten alternative labels with the highest number of distinct sources were annotated.
7. Roots matching rewritten preferred labels were annotated.
8. Roots matching rewritten alternative labels with the highest number of distinct sources were annotated.

The longest non-nested annotations for a given text chunk with the lowest priority value were retained and all the other annotations were deleted.

Priority	Preferred	Alternative	Rewritten	Root
1	✓			
2		✓		
3	✓			✓
4		✓		✓
5	✓		✓	
6		✓	✓	
7	✓		✓	✓
8		✓	✓	✓

Table 3: Priorities for label disambiguation on the gazetteer level in descending order. Columns 2 to 4 describe origins of the label.

We prepared two kinds of gazetteers – Drug gazetteer gathering data from DrugBank and Disease gazetteer that contained UMLS concepts from 14 biomedical semantic types related to diseases and body parts (T019, T020, T022, T023, T029, T030, T046, T047, T048, T049, T050, T184, T190, T191). The gazetteers were used for NER within the predefined sections only and the combination of section and gazetteer was used to define the annotation groups listed in Table 4.

Semantic section name	Used Gazetteer	Interpretation
Proprietary Drug Name	Drug	about
Indication	Disease	has indication
Other Drug Names	Drug	about
Drug Interactions	Drug	interacts with
Adverse Events	Disease	has adverse event
Contradictions	Disease	has contraindication

Table 4: The semantic sections with the Gazetteers applied and the interpretation of annotations, i.e. the annotation groups defined in the template.

Our approach to performing disambiguation has several advantages to methods described in Section 2. First, removing the label ambiguity at the stage of populating gazetteer dictionaries has a huge performance impact, as it needs only to be done once during initialization. In contrast, having any rule- or ML-based analysis performed for each annotation will definitely slow down the process and might not be feasible for large corpuses. Second, the approach is deterministic, which means that applying the same set structured knowledge over the same text will always produce the same results. Therefore, it is easy do detect why problems occur and correct them. The trade-off is that in this way

we gain precision and sacrifice recall. However, we consider the trade-off beneficial as we aimed at better precision from the beginning and expect the evaluation, described in Section 6, to confirm that.

4.4 Linking with structured knowledge

During the NER, several attributes are specified for each annotation. First, the *resource* tag is set, the value of which is an instance URI from the semantic repository, used for populating the gazetteers. Second, a *rel* attribute is set, which describes the relation of the instance to the section/document. The value assigned is a predicate URI, constructed from an application specific namespace prefix and a local name derived from the meta-model. The prefix is the domain name of the application serving the meta-model. The knowledge category associated with the section for the particular gazetteer, which made the annotation, defines the local name. Determining the relations is what we call interpretation. A summary of the annotation schema is given in Table 5.

Annotation Type	Features
section	• cleanChapter – the section title • tocNumber – the section number (if exists) • level – the section level (e.g. level=1 for tocNumber=3, level=2 for tocNumber=3.1, etc. If there is no tocNumber, level=1.) • section_ids – an array of matching semantic section names for all defined document templates • ID – the semantic section name after the document classification • typeof – the URI of the semantic section for the document template • resource – the URI of the semantic section for the specific document
lookup	• gap – the GAPP name, which created the annotation • section – the semantic section name • level – the section level • rel – the URI of the annotation, defined for the combination of the semantic section and GAPP that was run over it • resource – the instance URI from the semantic repository • string – the annotation string • typeof – the class URI from the semantic repository

Table 5: The annotation scheme used for both classification and NER.

The RDF statements are generated according to the scheme <*section*> <*rel*> <*resource*>. Finally, all statements are added back to the semantic repository (the KB). Inference rules and additional indexing is applied at this step. The knowledge categories hierarchy is also transformed to RDF in a similar manner, allowing for rich semantic queries over the document data using *skos:broader* and *skos:narrower* properties.

4.5 Generic disambiguation of UMLS terms

The initial semantic annotation pipeline reported in Khresmoi deliverable D1.2 [12] did not handle term ambiguity and retained all interpretations of a term found in UMLS. When manual correction of the output of this application was started, as described in Khresmoi deliverable D1.3 [13], simple heuristics were implemented to decrease the manual correction burden. They were implemented and delivered in the second prototype, as described in Khresmoi deliverable D1.4.2 [2]. This method formed the basis of disambiguation in these early prototypes, and is elaborated below.

The disambiguation heuristics consists of three steps:

1. If a given span of text is covered by several terms from UMLS, then only the longest will be retained. The intuition here is that UMLS contains terms for both atomic concepts and more complex, compound terms. We are interested in the compound terms as the atomic concepts may be derived from these at a later stage.
2. If a given span is covered by several terms of the same length (after the previous step), then any that are not considered "preferred" terms by UMLS are rejected. Each concept in UMLS may be described by multiple terms. One of these will be considered the preferred term, i.e. the one that is in common currency. The intuition here is that if multiple concepts map to a span of text, the most likely concept being described is the one that is generally used in natural language.
3. If several terms remain spanning the text, then one is selected based on the heuristic used by Cengage Learning, as described in [14]. This heuristic makes use of the concept identifier assigned to each concept in UMLS, the CUI. The CUI has a numeric portion. Although not designed to contain meaning, Cengage shows that the lower the numeric portion of a CUI, the more likely it is to refer to the common usage of a concept. This could be because the common usage is more general, and when CUIs are assigned to portions of the UMLS, they are assigned linearly.

Some initial evaluations of this method were reported in [13]. For future iterations of the application, we intend to combine (or replace as appropriate) this work with the knowledge based work on disambiguation reported in this deliverable.

5 Khresmoi Knowledge Base interface

As described in Section 4, the methodology proposed depends on the semantic sectioning. This sectioning is not generic in the sense that it can be used *as-is*, but requires user input – namely to define the document meta-model. In order to make this process as easy and flexible as possible Ontotext developed a web interface and processing infrastructure.

Users with administrative permissions can use a web UI to define the document meta-model(s). For this task we developed, in the course of several iterations, a view that enables the definition of tree-like section structures, selection of extraction resources per section, and their linking to semantic knowledge categories for interpretation. See Figure 2 for a screenshot of the interface.

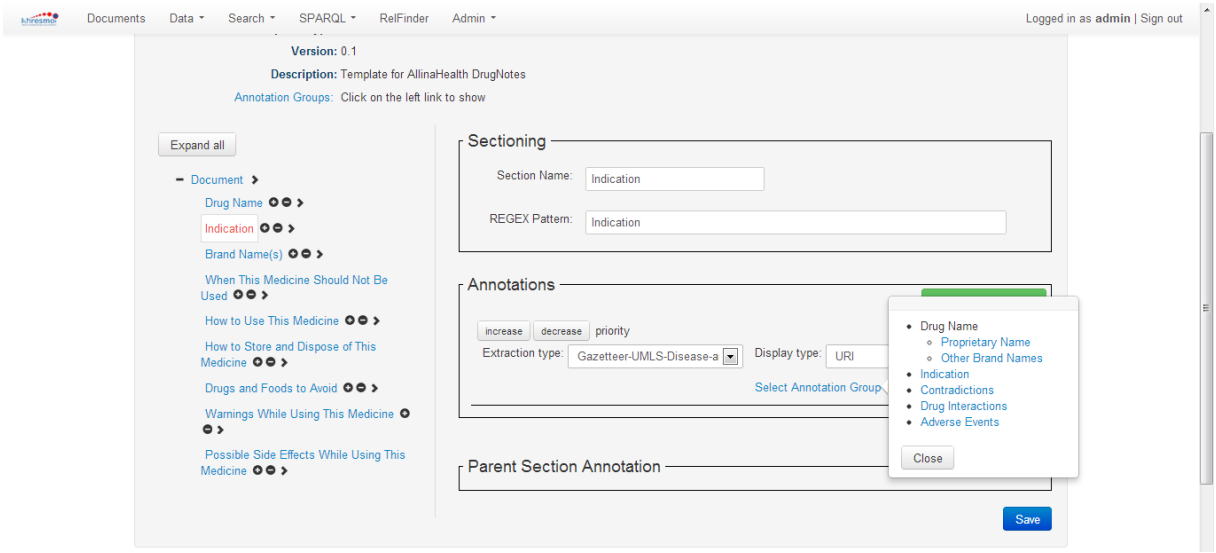

Figure 2 : Interface for a meta-model definition. The screenshot shows the definition of a section called *Indication,* which will be annotated with UMLS diseases and organs. The available semantic knowledge categories are also visible.

The subsequent process of performing semantic information extraction is divided into four steps:

- **Data upload and transformation:** the documents are uploaded to the system and, if necessary, transformed to html. Supported formats for conversion are MS WORD and PDF.
- **Sectioning**: syntactic and semantic sectioning is performed. Finally, a class is assigned to the document from the predefined templates.
- **Annotation:** semantic named entity recognition is performed on each semantic section with the corresponding extraction types from the meta-model.
- **RDFisation:** the semantic sections and their relationships to the extracted entities are serialised as RDF and are persistent in the semantic repository.

The steps have to be performed in the given order, but users can re-run the process from any stage if the meta-model is modified. All of the steps are exposed as RESTful web services, allowing the staging of batch document processes not only from the UI, but also with external tools and scripts. A summary of the Web API is given in Table 6.

Step	Relative URI	HTTP Method	Parameters		
Upload and transform	/transform	POST	uri	URI	Required
			title	String	Required
			file	Multipart file	
			url	URL	
Classify (with sectioning)	/section	GET	uri	URI	Required
NER	/annotate	GET	uri	URI	Required
Linking with structured knowledge	/rdfize	GET	uri	URI	Required

Table 6: REST API implemented for the whole semantic annotation approach described in this deliverable.

There is also a web presentation of the annotated documents. The document view renders the HTML as close as possible to the original form of the document. Document properties are displayed in a summary box. In addition, the semantic sections tree of each document is made available for navigation. Finally, the extracted entities are also displayed, grouped by categories from the document meta-model. Navigation to instances of each entity in the text is also implemented. An example of the view is shown in Figure 3.

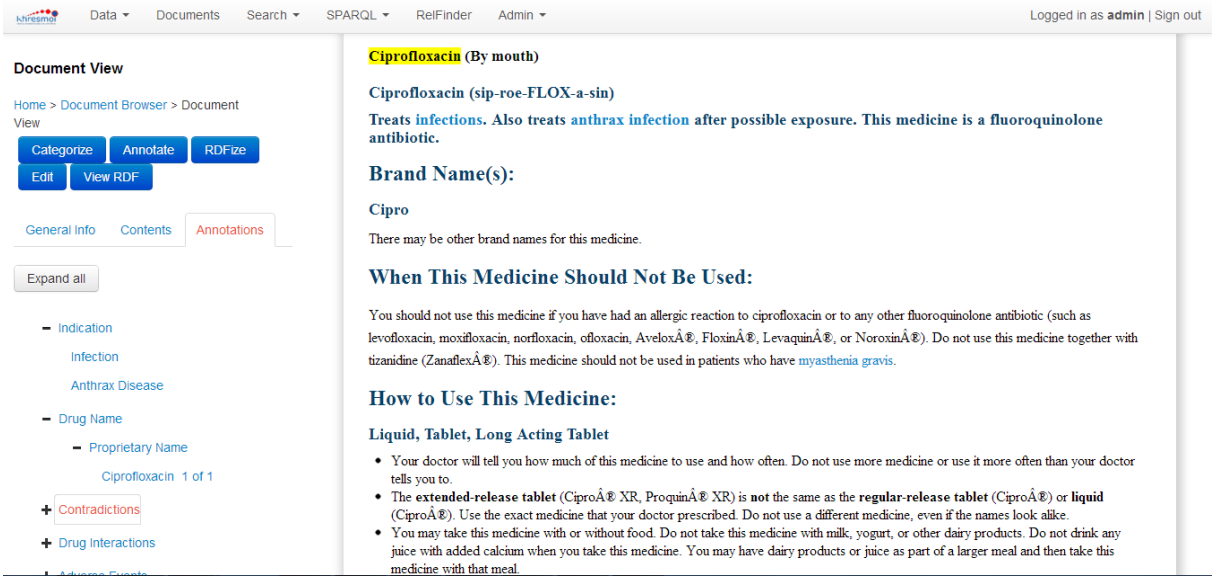

Figure 3 : Web view of an annotated document. The selected drug (Ciprofloxacin) occurrence is highlighted in the text. The left hand side panel displays content and extracted information with navigation capabilities.

6 Results and Evaluation

As previously mentioned we wanted to evaluate the results of the semantic annotation and disambiguation approach by comparing the annotation types assigned by our approach to a corpus of manually annotated documents. In this section, we present and analyse the results of this comparison.

A gold standard corpus was created for this evaluation by manually annotating 400 (~20%) randomly selected documents from our initial document set, described in Section 3. This manual annotation was conducted by the Lighthouse subcontractors using the procedure described in D1.4.2 [2]. Each document was processed by five annotators. The consensus between the annotators was automatically generated by selecting the annotation placed by the majority of the people. We compared the annotations from the gold standard set to the annotations of the corresponding document set. The metrics we derived are as usual the percentage of correct annotations (precision), the fraction of retrieved annotations (recall) and the F-score. The following formulas were used for the calculations:

$P = {A_{semantic} \cap A_{manual}}/{A_{semantic}}$, where $A_{semantic}$ and A_{manual} denote the annotation sets for the semantic annotation with disambiguation and the manual annotation respectively and P is the precision

$R = {A_{semantic} \cup A_{manual}}/{A_{manual}}$, where $A_{semantic}$ and A_{manual} denote the annotation sets for the semantic annotation with disambiguation and the manual annotation respectively and R is the recall

$F = 2\ {P\ *\ R}/{P + R}$, where P and R are the precision and recall respectively

The results summary is presented in Table 7.

Documents	400
Annotators	15
Annotators/document	5
Annotations	3465
Precision (strict)	0.97
Recall (strict)	0.62
F (strict)	0.76

Table 7 : Safe consensus overall scores.

From the values in Table 7 two observations stand out – a very high precision score and a comparably low recall. While the approach was aiming by design at good precision, which according to the results was achieved, it is still worth investigating the reasons for this performance.

There are several issues, which cause the recall score to be low:

1. Non-content annotations. Some manual annotations were created in non-content HTML tags such as <head> or the navigational frames of the web page.
2. The reason for most of the manual annotations missed by the automatic processing was the preliminary limitations described in Table 4. We had decided that finding different annotation types should be limited to specific semantic sections only. For example, the drug names finding was restricted to the semantic sections, which described:
    - The drug name for which the drug note was written (let's name it main drug).
    - The drug name that should not be used in conjunction with the main drug.
3. During the evaluation process we discovered that drug names finding was overlooked for "This Medicine Should Not Be Used" section (defined as "Contradictions" semantic section in Table 4) by the automatic processing, so the omitted drug names in that section could be considered as true false negative results. On the other hand, the drug names occurrence in "How to Use This Medicine" section were deliberately neglected, as the semantics of the drug names mentioned there was ambiguous.
4. Some literals are not present in UMLS, e.g. blood flow disorders, pale stool, dark-colour urine, sleepiness, fast heartbeat.

Issues 1 and 2 above are the result of an oversight during the definition of the annotation guidelines – the valuable text sections had to be marked accordingly so the manual annotators could limit their work on them only. The false negative problem in Issue 3 above can be easily fixed by running the Drug gazetteer over the "Contradictions" section. The last issue can be resolved either by adding new data sources or by enriching the rewrite rules used for the generation of derivative labels. Future work will look at improving on these fronts.

Even though the precision score is very high, we still identified several problems that should and will be addressed in the future.

1. For instance "Blood" was wrongly determined as a Disease concept type because of the combination of two kinds of errors:
    - Presence of a low quality literal for leukaemia disease – "Blood (Leukaemia)". According to one of the rewrite rules used for the generation of derivative labels, the original literal was shortened by removing the string enclosed in the parenthesis. Thus, "blood" was generated as a type of disease.
    - The disambiguation mechanism priorities (see Table 3) were designed to diminish the problems caused by the presence of low quality literals. The highest priority (1) of "blood" had to be set for the concept of type Tissue if all UMLS semantic types were used. However, the Tissue concept type (together with many other concept types that were out of the project scope) was removed as a first step of the Disease gazetteer population.
2. Similarly, "Drug" was wrongly determined as a Disease concept type because of the combination of two kinds of errors:
    - The root of the symptom "drugged" is "drug".
    - Similarly to the case with the "blood" literal, the disambiguation priory mechanism had to place the correct concept type (Pharmacologic Substance) to the "drug" literal (with priority 2) instead of Disease if all UMLS semantic types were used as an input at that step.

Both low quality literals and erroneous roots finding problems can be resolved to a great extent by using all UMLS concept types for the gazetteer population and, as a final step, filtering of the concept types that are irrelevant for the project.

In conclusion, the results confirm that the presented approach fulfils the goals set – performing semantic information extraction with a novel high precision disambiguation method that does not rely on the existence of gold standard/training corpuses. The results derived from the semantic annotations can be used to both enhance the search functionalities of the system by providing specific interpretations of the identified named entities and for enriching the knowledge in the KB.

7 Conclusion

In this document we presented a novel approach for performing semantic NER with reference disambiguation that is based on pre-defined document meta-models. The implementation of the methodology and its application on a corpus of bio-medical web pages successfully linked terms from the documents to entities in the KB with high precision depending on their semantic interpretation. Because the presented approach is not applicable to new document types that don't have a defined template, we also present a complementary disambiguation strategy that is in line with the general semantic NER approach described in D1.4.2 [2]. In future work we will aim to use both IE approaches together in order to improve the quality of the search results retrieved by the Khresmoi system. This will involve work on identifying content source of particular interest for which to define the document meta-models.

8 References

[1] Roberts A., Petrak J., Aswani N. Report accompanying Manually Annotated Reference Corpus. Khresmoi project deliverable D1.4.1. May 2013

[2] Roberts A., Petrak J., Aswani N. Report on Coupling Manual and Automatic Annotation. Khresmoi project deliverable D1.4.2. May 2013

[3] Hanbury A., Boyer C., Gschwandtner M., Müller H. KHRESMOI: towards a multi-lingual search and access system for biomedical infromation. Med-e-Tel, Luxembourg, 2011.

[4] Kiryakov A., Davies J. Semantic Search. "Information Retrieval - Searching in the 21st Century"; Goker A. (Editor), Davies J. (Co-Editor), Graham M. (Co-Editor). John Wiley & Sons, Europe, 2007

[5] K. Pentchev, V. Momtchev. D5.1 Report on data source integration

[6] K. Pentchev, V. Momtchev. D5.2 Large Scale Biomedical Knowledge Server

[7] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer. Word-sense disambiguation using statistical methods. Proc. 29th ACL, Berkley, California, 1991

[8] V. Hatzivassiloglou, P. A. Duboue, A. Rzhetsky. Disambiguating proteins, genes, and RNA in text: a machine learning approach. Bioinformatics, 17:97–106, 2001

[9] R. Bunescu, M. Pasca. Using Encyclopedic Knowledge for Named Entity Disambiguation. EACL. Vol. 6. 2006

[10] X. Han, J. Zhao. Named Entity Disambiguation by Leveraging Wikipedia Semantic Knowledge. Proceedings of the 18th ACM conference on Information and knowledge management. ACM, 2009

[11] H. Cunningham, et al. Text Processing with GATE (Version 6). University of Sheffield Department of Computer Science. 15 April 2011. ISBN 0956599311.

[12] M. A. Greenwood, A. Roberts, N. Aswani, Ph. Gooch. Initial prototype for semantic annotation of the Khresmoi literature. Khresmoi project deliverable D1.2. May 2012

[13] N. Aswani, L. Kelly, M.A. Greenwood, A. Roberts, M. Samwald, N. Pletneva, G. Jones, L. Goeuriot. Report on results of the WP1 first evaluation phase. Khresmoi project deliverable D1.3. August 2012

[14] King, B., Wang, L., et al. (2011). Cenagage Learning at TREC 2011 Medical Track. The Twentieth Text Retrieval Conference Proceedings (TREC 2011), Gaithersburg, MD. National Institute for Standards and Technology.