RENDER
FP7-ICT-2009-5
Contract no.: 257790
www.render-project.eu

RENDER

Deliverable D2.2.2

Final Version of the Fact Mining Toolkit

Editor:	Delia Rusu, JSI
Author(s):	Delia Rusu, JSI; Mitja Trampus, JSI; Inna Novalija, JSI; Tadej Stajner, JSI; Mariana Damova, Ontotext
Deliverable Nature:	Prototype (P)
Dissemination Level: (Confidentiality)	Public (PU)
Contractual Delivery Date:	31 March 2012
Actual Delivery Date:	31 March 2012
Suggested Readers:	Developers working on WP4 – Diversity Toolkit, developers creating case study prototypes (WP5)
Version:	2.0
Keywords:	Fact mining, article template discovery, integration, evaluation, RDF

Disclaimer

Full Project Title:	RENDER – Reflecting Knowledge Diversity
Short Project Title:	RENDER
Number and Title of Work package:	WP2 Diversity Mining
Document Title:	D2.2.2 - Final Version of the Fact Mining Toolkit
Editor (Name,Affiliation)	Delia Rusu, JSI
Work package Leader (Name, affiliation)	Delia Rusu, JSI
Estimation of PM spent on the deliverable:	8

Executive Summary

This deliverable presents the final version of the Fact Mining Toolkit, and it is a continuation of the Prototype of the Fact Mining Toolkit deliverable.

We start by describing the RENDER general architecture as a three-tier architecture with the Fact Mining Toolkit belonging to the Application Layer.

Furthermore, we describe how the Fact Mining Toolkit is integrated in the RENDER architecture via the Enrycher service-oriented system which exposes the functionality of the Fact Mining and Opinion Mining Toolkits using REST web services. We detail the integration of the Fact Mining Toolkit with respect to the Knowledge Diversity Ontology and ontologies from the Reference Knowledge Stack (mainly PROTON) which are used to represent the extracted textual information in RDF. We also describe more in depth the knowledge infrastructure and the faceted search functionality.

Next, we evaluate the Fact Mining Toolkit in terms of the informativeness and correctness of the extracted facts. Our findings show that the majority of assertions extracted using our methods can be considered informative and therefore sufficiently useful in indexing for Information Retrieval.

In the final part of the deliverable we present the article template discovery toolkit component. We focus on the differences with respect to the initial approach to this task, describe the algorithm and continue with its evaluation and a few illustrative examples.

List of authors

Organisation	Author
JSI	Delia Rusu
JSI	Mitja Trampus
JSI	Inna Novalija
JSI	Tadej Stajner
Ontotext	Mariana Damova

Table of Contents

List of Figures

List of Tables

Abbreviations

KDO	Knowledge Diversity Ontology
RDF	Resource Description Framework
XML	Extensible Markup Language
RKS	Reference Knowledge Stack
SIOC	Semantically-Interlinked Online Communities
DMOZ	Open Directory Project
GATE	General Architecture for Text Engineering

Definitions

OWLIM is a family of semantic repositories, or RDF database management systems.

SIOC (Semantically-Interlinked Online Communities) Core Ontology is an ontology which provides the main concepts and properties which allow to describe information from online communities.

Blank node (or bnode) In an RDF graph, a bnode represents a resource which has no given URI or literal; such a resource is also called an anonymous resource.

PROTON (PROToONtology) is an ontology developed in the SEKT project as a lightweight upper-level ontology.

GATE (General Architecture for Text Engineering) is an open source text processing software.

1 Introduction

The Diversity Mining work package is mainly focused on discovering diversity from structured and unstructured datasets, such as mainstream news and social media. We mine information diversity by identifying facts and opinions from text, and representing them in a structured form (RDF) for further inference and querying.

This deliverable presents the final version of the Fact Mining Toolkit, and it is a continuation of the Fact Mining Toolkit prototype. In the Fact Mining Toolkit prototype deliverable [23]we described a Fact Extraction System providing shallow as well as deep text processing functionality at the text document level. Among the features of the system, we described topic and keyword detection and named entity extraction as shallow text processing features, and named entity resolution and merging, word sense disambiguation and assertion extraction as deep text processing features. Moreover, we described an example application of Enrycher for news fact extraction and presented on-going research work on article template discovery.

The final version of the Fact Mining Toolkit presents the integration of the fact mining tools within the RENDER component architecture, and specifically focuses on the integration with the Knowledge Diversity Ontology (KDO) developed in WP3 and the RDF repositories described in WP1. We also describe more in depth the knowledge infrastructure and the faceted search functionality.

Furthermore, the deliverable describes the evaluation of the Fact Mining Toolkit in terms of the informativeness and the correctness of the extracted facts. Finally, we describe a new version of the article template discovery algorithm which was presented as on-going research work in the prototype deliverable.

The deliverable is structured as follows. We start by describing the RENDER architecture overview in Section 2, followed by the Fact Mining Toolkit integration in Section 3 – where we focus on the integration with the Knowledge Diversity Ontology (KDO) and OWLIM and the Reference Knowledge Stack (RKS). Section 4 describes the knowledge infrastructure and faceted search. The Fact Mining Toolkit evaluation is presented in Section 5, while Section 6 describes the article template discovery algorithm. The final section of the deliverable presents concluding remarks and future work.

2 RENDER Architecture Overview

In order to better understand the integration of the Fact Mining Toolkit, we start by briefly presenting the RENDER architecture overview together with its main components (see Figure 1).

The RENDER architecture can be seen as a classical three-tier architecture, with a data layer which covers data collection, data storing as well as provides data querying functionality. At the level of the work packages, this is related to the work described in WP1 – Data collection and management and WP3 – Diversity representation and processing.

The following layer, the application layer, includes the main tools developed within the project. These include the Diversity Mining Services (Enrycher) comprising the Opinion Mining and Fact Mining toolkits developed in WP2 – Diversity mining, as well as the CLAS toolkit for diversity ranking developed in WP3 - Diversity representation and processing.

The final layer, the presentation layer, includes three dashboards developed by the case study partners in WP5 – Diversity case studies, as well as extensions of Drupal, Media Wiki and Semantic Media Wiki. In addition, we also develop a series of demo websites which expose parts of the functionality available in the application layer: the Enrycher and Opinion Mining demos, as well as a Search over sentiments and entities demo.

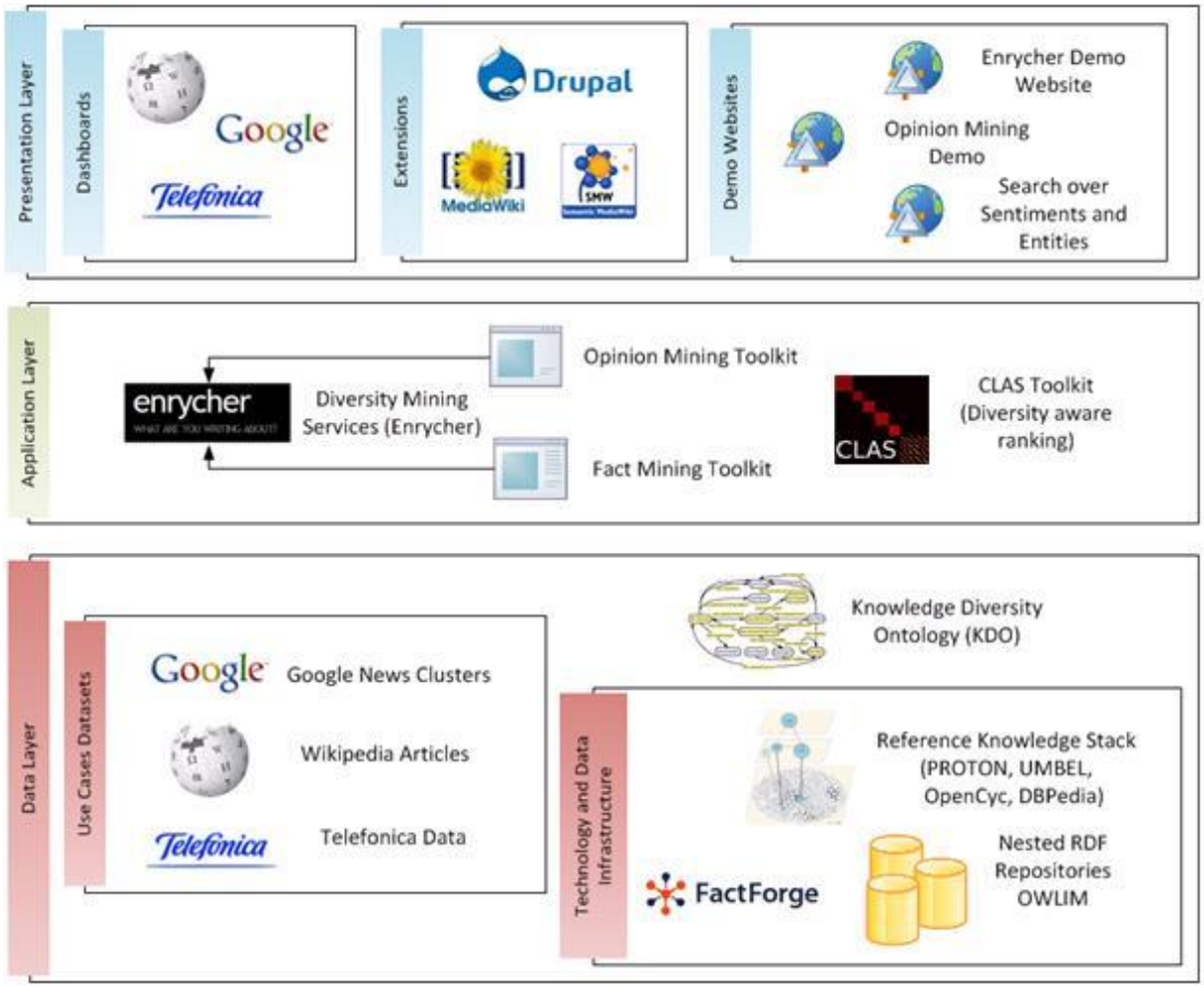

Figure 1.The RENDER Architecture overview.

Note that this is the current version of the RENDER architecture, with more components to be added and integrated in the second and third years of the project.

3 Fact Mining Toolkit Integration

The Fact Mining Toolkit is one of the main application layer components of the RENDER project (see Figure 1). The functionality provided by both the Fact Mining Toolkit as well as the Opinion Mining Toolkit components was exposed as a set of services and integrated within the Diversity Mining suite of services (Enrycher).

Within the RENDER architecture, the Fact Mining Toolkit is responsible for processing raw data provided at the data layer, for e.g. Wikipedia articles, news articles or blog entries. The components belonging to the toolkit extract and resolve various types of information: categories, entities, disambiguated words, subject-predicate-object assertions, article templates. The output of each of these components is made available to the other architecture layers via the Diversity Mining services (also referred to as Enrycher).

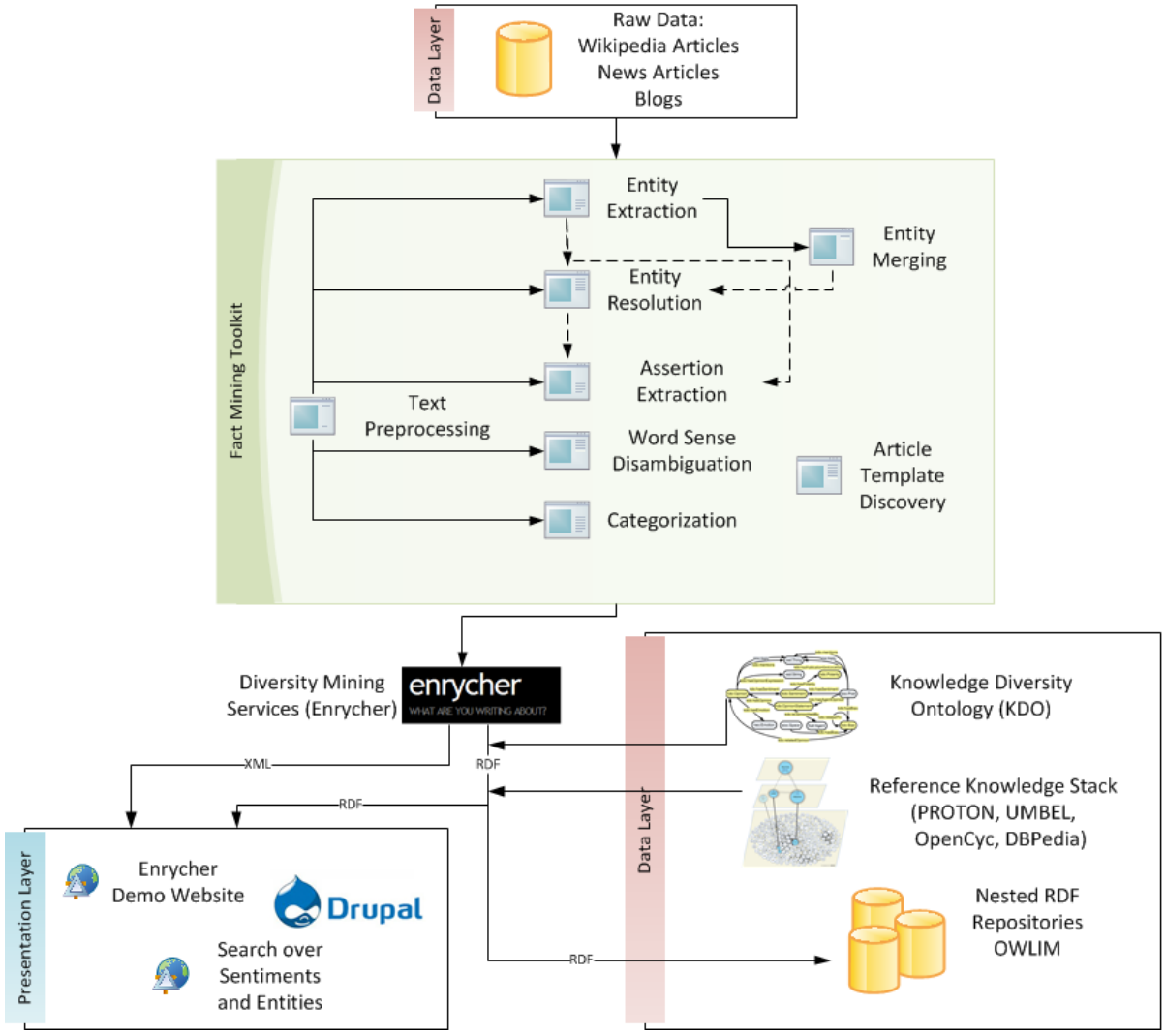

Figure 2.The integration of the Fact Mining Toolkit within the RENDER architecture.

The Enrycher services accept two types of output: either XML of RDF representations. The RDF representation of the Enrycher services is conformant to the description of the Knowledge Diversity Ontology (KDO) developed within RENDER, as well as the ontologies included in the Reference Knowledge Stack (RKS)[24]. The following two sub-sections will further detail the integration between the Enrycher services and the KDO and RKS.

The client applications of Enrycher can request the output of individual components (for e.g. extracted and resolved entities) or combine several components in a pipeline, as described in [23]. Currently there are several client applications which rely on components from the Fact Mining Toolkit: the Enrycher demo website, the Drupal extension developed in WP4 – Diversity Toolkit, and the Search over sentiments and entities demo based on FactForge[6]. Future tools, for e.g. from the Wikipedia Dashboard, will also rely on a number of Fact Mining Toolkit components.

3.1 Integration with the KDO

The Knowledge Diversity Ontology (KDO) [26]was developed with the purpose of describing the textual information that is extracted, processed, stored and retrieved by the RENDER software components. Thus the KDO facilitates the communication between different software components developed and maintained by different project partners. In this sub-section we detail the integration between the KDO and the Enrycher service-oriented system.

The integration was done at the level of the RDF representation that Enrycher outputs, as follows:

1. **The text to be processed with the Enrycher pipeline of services.** Within RENDER we are dealing with several types of textual information: news articles, blogs, Wikipedia articles, tweets, etc. The most generic way of representing the type of textual input is **sioc:Post**(defined in the SIOC ontology[16][27]). If the information to be processed comes along with additional metadata which contains the type of the information (e.g. news article, blog post, etc.) we can utilize other more specific classes. For representing the "news article" KDO has a specific class **kdo:NewsArticle**. The content of the textual information to be processed using Enrycher (the text body) is represented via **sioc:content**.
2. **Information extracted from DMOZ**[2]**.** The Enrycher Categorizer service extracts topics and keywords based on the DMOZ hierarchy of concepts. The topics are represented using **sioc:topic** property, whereas the keywords are represented with the **sioc:tag**property.
3. **Telefonicataxonomy of concepts.** We have created a taxonomyof concepts for the Telefonica specific topics and sub-topics[20].
4. **The annotations.** The annotations (and similarly, sentiments) are represented as blank nodes (see the Definition section for an explanation of blank nodes). This is because we are extracting textual information which is not sure to have an equivalent RDF concept. After we extract the textual information and assign it an anonymous resource, we try to resolve this information either via the Entity Resolution or Word Sense Disambiguation services.

3.2 Integration with OWLIM and RKS

The textual data processed with Enrycher is stored in the RDF family of semantic repositories OWLIM[13].In order to facilitate the integration between the Enrycher RDF output and the prerequisites of the OWLIM semantic repositories, several changes have been performed at the level of the RDF representation of annotations based on the PROTON [6] ontology of the Reference Knowledge Stack (RKS):

- The start and end index of the annotations in text is stored using the **pkm:startOffset** and **pkm:endOffset** properties of the PROTON ontology.
- The annotations disambiguated with WordNet concepts are stored using the **dbp-prop:wordnet_type**DBpedia property.
- The entities extracted using the GATE[29]application are mapped to PROTON concepts, as shown in Table 1.

Table 1. Mapping between GATE entities and PROTON concepts

GATE entities	PROTON classes
Person	ptop:Person
Organization	ptop:Organization
Location	ptop:Location
Date	pext:Date
Time	ptop:TimeInterval
Money	pext:Currency
Percentage	pext:Percent
Male	pext:Man
Female	pext:Woman
City	pext:City
Country	pext:Country
Region	pext:GeographicRegion

3.3 Integration Tool

In order to automatically process streamed textual data, provided either by the Spinn3r client application[23]or the news crawling system[25](both developed by JSI), JSI and Ontotext have been working on an Integration Tool. The purpose of this tool is to receive a text item (a news article, blog), call the Enrycher pipeline of services and send the resulting RDF representation to the OWLIM repositories for storing. The tool is currently being tested on news articles provided by the JSI Spinn3r client, and will be made available to the consortium in the following months. As future work we are going to extend the tool in order to automatically process Wikipedia articles as well.

4 Knowledge Infrastructure and Faceted Search

4.1 Data Layer Infrastructure

This section describes the integration of the data produced by the fact mining tool into RENDER data layer infrastructure.
The RENDER data layer infrastructure stores and manages data represented according to the following vocabularies and ontologies:

(a) the RKS (the reference knowledge stack) introduced in deliverables 1.1.1 and 1.2.1 and presented in [1], and the reason-able view [8], [11] – FactForge[6], on top of which the RKS is a layer. It provides efficient access to the heterogeneous data from a segment of LOD, that are part of FactForge (see deliverables 1.1.1, 1.2.1, and 1.3.2). The version of the RKS presented in this deliverable features PROTON ontology [14], [15] mapped to DBpedia ontology [3], Geonames ontology [8] and Freebase [7].

(b) KDO (Knolwedge Diversity Ontology) developed in WP3 of RENDER project [12]

(c) External sources such as SIOC (Semantically-Interlinked Online Communities) [16], and HEO (Human Emotions Ontology)[9], dc (Dublin core) [4], etc.

The RKS allows users to formulate SPARQL queries with PROTON predicates only, and obtain results from the entire LOD segment in FactForge. For example, the query about Software companies founded in the US contains only PROTON predicates, and returns 30 results of companies located in California, Colorado, Missouri, Florida, Oregon, etc. (see Figure 3 below).

SELECT DISTINCT ?Company ?Location

WHERE {

?Company**rdf**:type**pext**:Company;
pext:industryOf**dbpedia**:Computer_software ;

ptop:establishedIn ?Location .

?Location**ptop**:subRegionOf**dbpedia**:United_States .

}

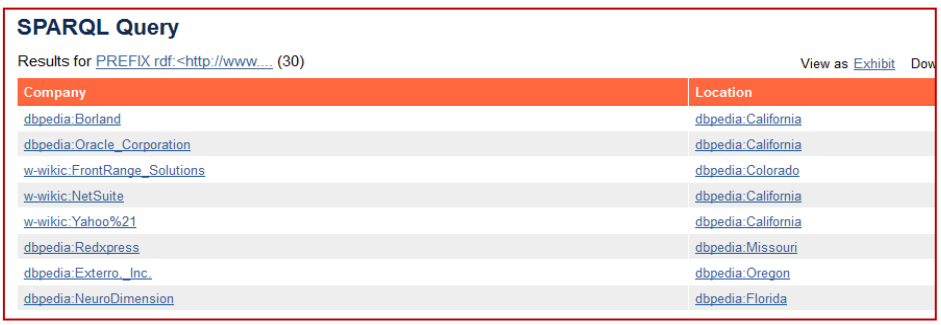

Figure 3. SPARQL query results for software companies founded in the US.

As it was outlined in section 3.2 the Enrycher fact mining tool uses PROTON ontology and KDO to represent the entities recognized in the documents it processes. Figure 4 shows the RDF model, according to which

the data produced by Enrycher data mining tool are represented. The annotation in this model is interpreted as aLexicalResource, Mention, defined in PROTON ontology. It is a blank node with the following links: (a) the annotation is mentioned by a document, (b) it refers to a DBpedia instance, which has a PROTON class. Additionally, the annotation has a subject (dc:subject) described by a DBpedia category, a wordnetsynset (dbp-prop:wordnet_type), offsets recording its position in the given document, and a string which is actual utterance describing the named entity in the text. The document is a News Article in KDO's terms and an Information Resource in PROTON's terms. It is represented with its body, e.g. the content of the document in text format, with topics according to SIOC, and DMOZ [2] or Telefonica, and finally with the annotations of the names entities recognized in it.

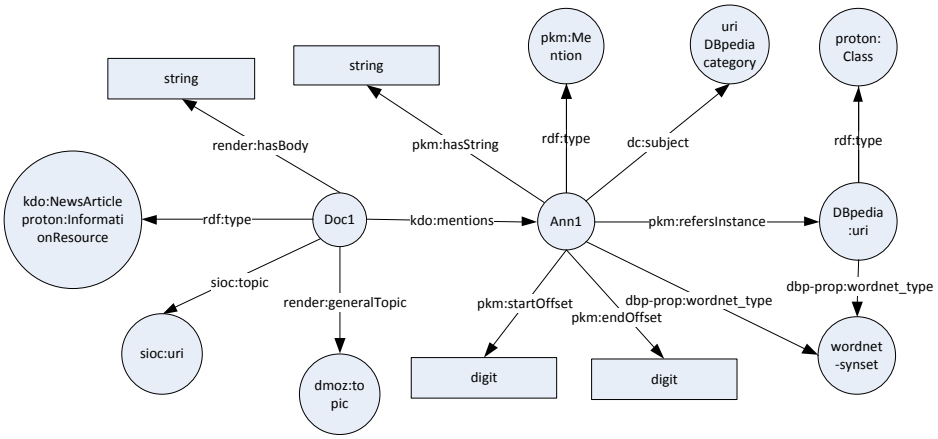

Figure 4.The RDF model used for representing data extracted by Enrycher.

The following 12 triples show an example of encoding in RDF of the information about one hypothetical annotation of the actor Brad Pitt in a hypothetical document:

1. <http://jsi.org/document-123>
   rdf:typekdo:NewsArticle , rdf:typeptop:InformationResource .

2. <http://jsi.org/document-123>
   sioc:topic<http://kdo.render-project.eu/topics/Telefonica/Movistar> .

3. <http://jsi.org/document-123>
   render:hasBody "document body as a string" .

4. <http://jsi.org/document-123>
   render:generalTopic<http://www.dmoz.org/Top/Arts> .

5. <http://jsi.org/document-123>
   kdo:mentions _:annotation0000 .

6. _:annotation0000 rdf:typepkm:Mention .

7. _:annotation0000 pkm:startOffset "10"^^xsd:integer .

8. _:annotation0000 pkm:endOffset "17"^^xsd:integer .

9. _:annotation0000 pkm:refersInstancedbpedia:Brad_Pitt .

10. _:annotation0000pkm:hasString "Brad Pitt" .

11. _:annotation0000
    dc:subject<http://dbpedia.org/resource/dbpedia/Category:American_film_actors> .

12. _:annotation0000 dbp-prop:wordnet_type wordnet:synset-stage-noun-4 .

The prototype of the integration of data produced by Enrycher fact mining tool into the RENDER data layer infrastructure includes 6605 documents, which amount to 1,129,238 explicit triples. Figure 5 shows that the number of retrievable triples with information from the documents is 30% bigger, e.g. the user disposes with 1,573,224 retrievable triples. These figures are obtained by subtracting of FactForge triples form the overall number of triples obtained after loading the data from the fact mining tool.

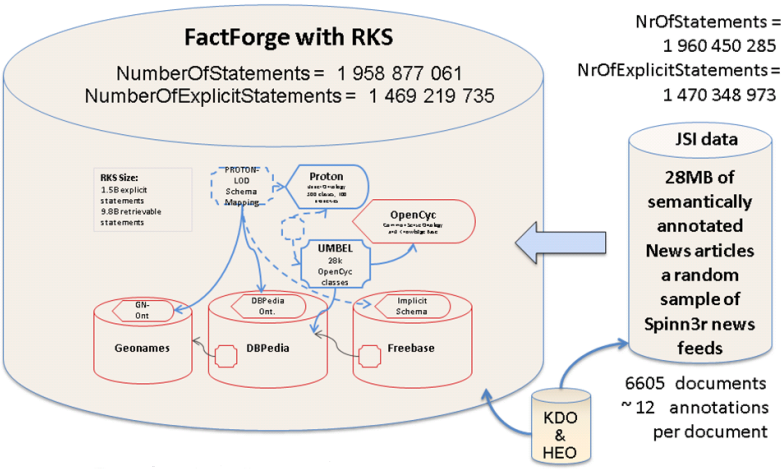

Figure 5. Quantification of information stored and processed at the data layer.

4.2 KIM Faceted Search

KIM is a platform for semantic annotation and multi-paradigm search over documents, data, and knowledge, developed at Ontotext. The present deliverable includes integration of the datalayer infrastructure described in the previous section with faceted search of KIM. “Faceted search, also called faceted navigation or faceted browsing, is a technique for accessing information organized according to a faceted classification system, allowing users to explore a collection of information by applying multiple filters … Facets correspond to properties of the information elements” [5]. Kim faceted search allows users to look for combinations of entities occurring together in documents. The version of KIM faceted search presented in this deliverable allows users to retrieve documents of particular topic that mentions particular entities. Figure 6 shows initial facets combining People, Locations, Organizations and Topics.

Figure 6.Initial facets combining People, Locations, Organizations and Topics.

Figure 7 shows the results of selecting the Location of Europe and the topic Health. The knowledge base returns 12 Health documents that mention a location in Europe along with some organizations and people. The snippets of the first two can be seen in the bottom of the screenshot of Figure 7.

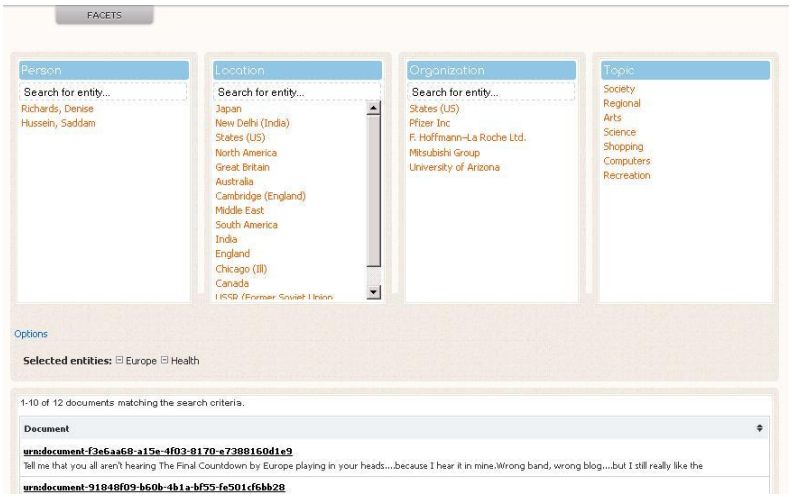

Figure 7.Results of selecting the Location of Europe and the topic Health.

This user interface connects with the data layer infrastructure described above through SPARQL queries over the RKS, and uses the integrated information produced from Enrycher fact mining tool to show the documents and based on the matching combinations of named entities and topics.

The faceted search capability of RENDER, presented in this deliverable, offers the users the possibility to select combinations of entities they are interested in to build their search facets by checking the concepts describing them in an intermediary interface, as shown on Figure 8. Note that this version of facets selection already includes Sentiment, and prepares the ground to build facets with combinations of named entities, but also with topics, opinions and sentiments, a functionality of the Search over Sentiments and Entities tool to be developed in the next stage of the project.

Figure 8.Facets options.

The selection of Figure 9 includes People, Organizations, and Topics. This interface allows to include sentiments and opinions in the facets. Thus, selection of documents will be possible based on combination of entities, sentiments and opinions, once the document information about sentiments and opinions will be available for use.

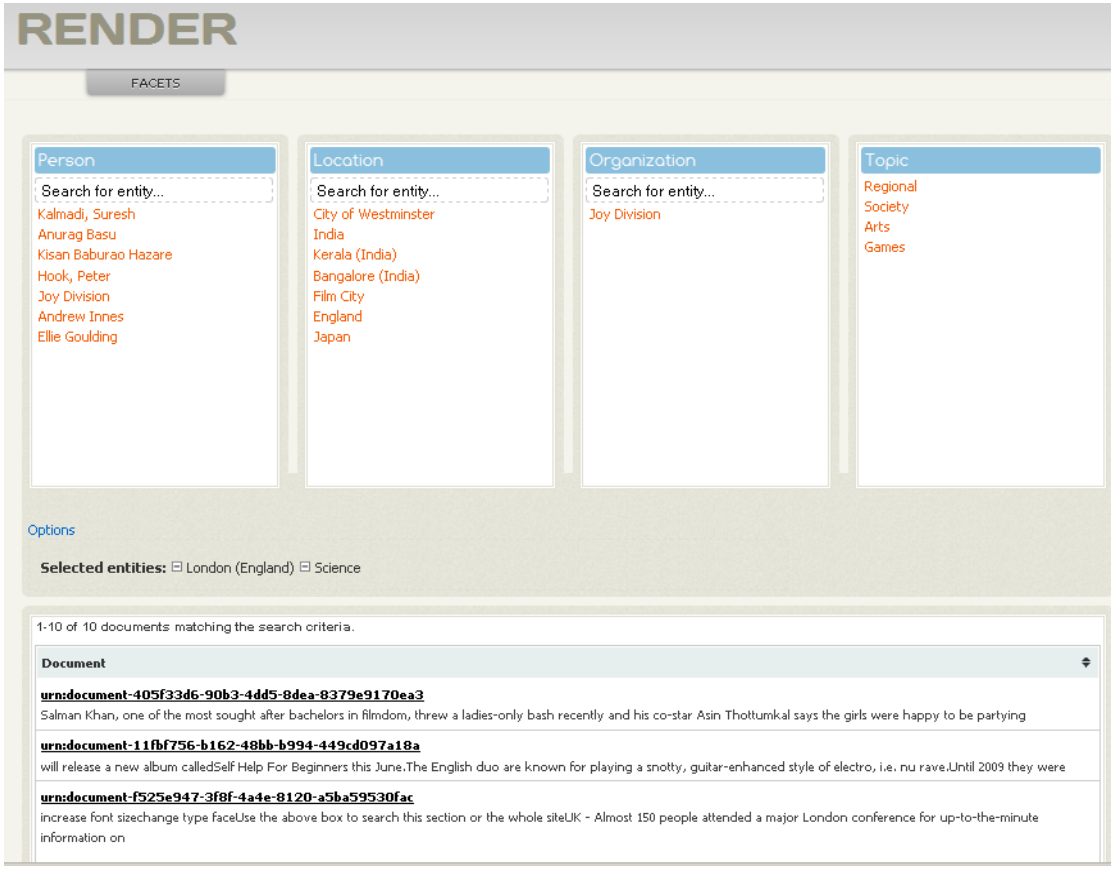

Figure 9. A selection including People, Organizations, Topics and Sentiment (empty).

5 Evaluation of the Fact Mining Toolkit

In the prototype deliverable of the Fact Mining Toolkit [23] we described the Fact Mining Toolkit components and discussed their evaluation, as it was performed for each component individually. In this deliverable we present an integrated evaluation where the aim is to analyse the performance of several combined components.

The evaluation of the Fact Mining Toolkit has been performed on a randomly selected set of news and Wikipedia articles. From the collection of 9GB of Wikipedia data and 3.36 GB of news data we have selected 100 news articles and 100 Wikipedia articles.Following that, we have pre-processed the selected articles and used the Fact Mining tools as exposed via the Enrycher[22]service-oriented system, providing shallow as well as deep text processing functionality at the text document level.

Enrycherincludes the following shallow processing functionality:

- topic and keyword detection;
- named entity extraction: names of people, locations and organizations, dates, percentages and money amounts.

The deep text processing Enrycher tasks include:

- named entity resolution with respect to existing Linked datasets: DBpedia[3], YAGO[18], OpenCyc[19];
- named entity merging: co-reference and anaphora resolution;
- word sense disambiguation into WordNet[17];
- assertion extraction, by identifying subject – predicate – object sentence elements together with their modifiers (adjectives, adverbs) and negations.

UsingEnrycher we have extracted 176 facts from news articles and 221 facts from Wikipediaarticles. Each fact has been extracted from a sentence in the article (note that a sentence can contain one or more facts). The facts that we extract consist of assertions: subject – predicate – object sentence elements together with their modifiers, as well as links to external resources (in our evaluation example DBpedia) that disambiguate the assertion elements.

For eachextracted fact we present the following information to the evaluators:
- the paragraph from news/Wiki article, where the fact occurs;
- the sentence from news/Wiki article, where the fact occurs;
- the subject from the fact assertion;
- the subject modifiers from the fact assertion;
- the subject DBpedia URI (if exists);
- the subject DBpedia abstract (if exists);
- the predicate from the fact assertion;
- the predicate modifiers;
- the object from the fact assertion;
- the object modifiers;
- the object DBpedia URI (if exists);
- the object DBpedia abstract (if exists).

Example 1 shows an extracted fact from a Wikipedia article. We show the paragraph of interest, the sentence containing the extracted fact (and belonging to the abovementioned paragraph), and the fact information as described above.

Example 1: Fact extracted from Wikipedia.

Paragraph: *USS Gerald R. Ford (CVN-78) 1] CVN-78 is to be the lead ship of her class of United States Navy supercarriers. When completed, she will be the first of the CVN-21 series of aircraft carriers. CVN-78 is currently scheduled to be laid down in 2009, concurrently or nearly so with the commissioning of USS George H. W. Bush (CVN-77). Construction work has already begun; on August 11, 2005, Northrop Grumman held a ceremonial steel cut for a 15-ton plate that will form part of a side shell unit of the carrier. If construction of the carrier remains on schedule the new ship should join the U.S. NavyOCOs active fleet as a fully commissioned warship sometime in 2015. Naming CVN-78 2007 National Defense Authorization Act: USS Gerald R. Ford On October 17, 2006, President George W. Bush signed into law the John Warner National Defense Authorization Act for Fiscal Year 2007. Section 1012 of the act declares that "it is the sense of Congress that the nuclear-powered aircraft carrier of the Navy designated as CVN-78 should be named the U.S.S...*

Sentence: *Naming CVN-78 2007 National Defense Authorization Act: USS Gerald R. Ford On October 17, 2006, President George W. Bush signed into law the John Warner National Defense Authorization Act for Fiscal Year 2007.*

Subject: *President George W. Bush*

Subject modifiers:

Subject DBpedia URI: *http://dbpedia.org/resource/President_George_W._Bush*

Subject DBpedia abstract: *The presidency of George W. Bush began on January 20, 2001, when he was inaugurated as the 43rd President of the United States of America. The oldest son of former president George H. W. Bush, George W. Bush was elected president in the 2000 general election, and became the second US president whose father had held the same office (John Quincy Adams was the first)...*

Predicate: *signed into*

Predicate modifiers:

Object: *law John Warner National Defense Authorization Act*

Object modifiers: *the*

Object DBpedia URI:

Object DBpedia abstract:

The crowdsourcing service **CrowdFlower**[21] was used for evaluating the extracted facts by several criteria. We have asked the evaluators two questions: firstly, how informative the provided facts are, and secondly, how correct the facts are. When measuring informativeness we took into account the assertions that define the fact, and aimed at determining if the assertions convey the same information as the sentences where they occurin or not. The perceived informativeness was evaluated on a Likertscale [30]1-5, where 1 is the least informative and 5 is the most informative with regard to the sentence in which it appeared. We have evaluated how correct the extracted facts are (on a binary scale) by determining if the assertion subject and object were correctly disambiguated to semantic concepts (in this case from DBpedia).

In case some information for a fact is missing – which can be the case for the DBpedia URIs and abstracts for the subject and object – the fact is still taken into account, and the evaluators are asked to assess only the informativenes of that fact.

Each example was judged three times by different annotators. Twenty-three examples were chosen as “gold” examples that were used to verify the annotators.

The results are as following: the average score assigned to the assertions is 4.103 , each item having mean annotation variance of 0.494. Furthermore, the scores assigned to assertions extracted from news articles is higher at 4.162, while Wikipedia-sourced triplets score at 4.058, a statistically significant difference (T-

test, p = 0.008). A possible reason for this is that the language in news articles tends to be more explicit and assertive than Wikipedia articles and also contain more named entities compared to their total length. The variances in these two subgroups are not significantly different.

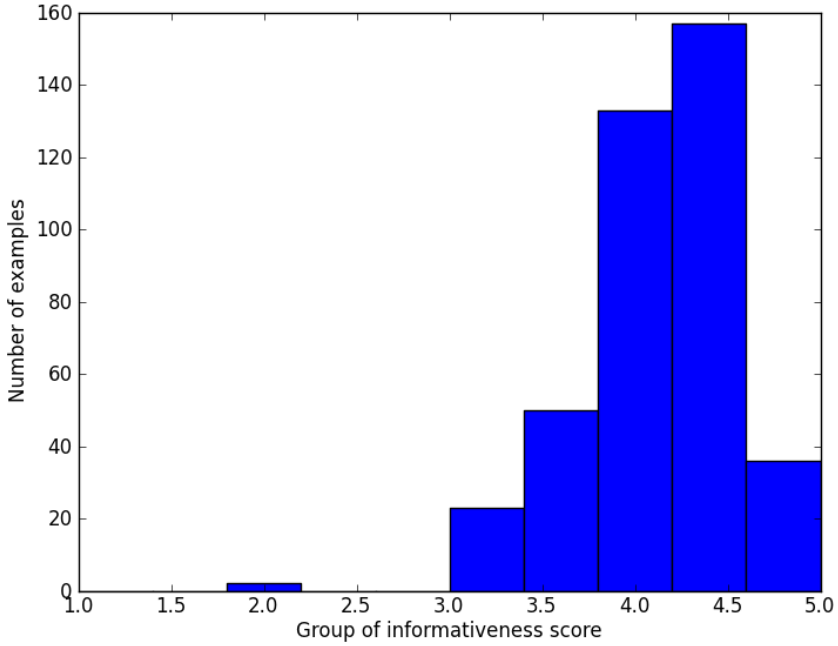

Figure 10. Histogram of average informativeness score, binned into 10 segments on the scale of 1.0-5.0

Figure 10 shows a bigger picture of the overall score distribution. While no examples are labelled with the worst possible score (1), very few examples (only two) are labelled with a 2 and the majority of the ratings fall between 3 and 5, while 23 have an rounded average score of 3, 321 examples of score 4 and 55 examples of score 5. This roughly means that while a quarter of the obtained assertions are equivalent to the sentence in terms of information, a large majority can still be deemed practically useful (having score 4) in some information retrieval applications.

Figure 11. Histogram of score variance, grouped by binned informativeness score. demonstrates that examples which have lower scores (2 or 3) also exhibit high score variances, meaning that annotators did not agree on their evaluations. This could be explained by some unreliability of the annotators, since they seem to prefer higher scores. High disagreement among annotators can also be interpreted as a misleading or ambiguous assertion, in which case we treat it as incorrect. Going further, we can also use this intuition of using variance to better estimate the proportion of useful triplets.

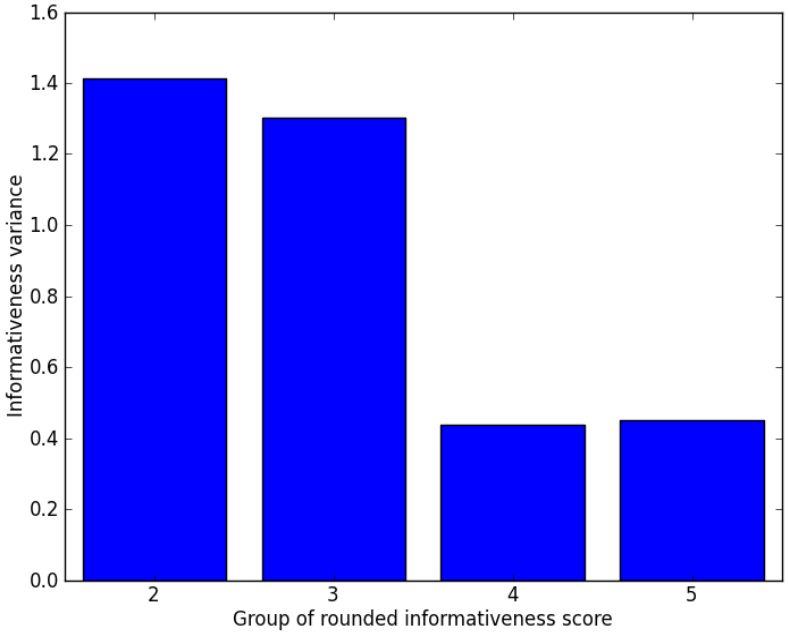

Figure 11. Histogram of score variance, grouped by binned informativeness score.

In order to compensate for the unreliability of annotators, we choose a stricter criterion for evaluation. Let the criteria *s* be the following: measure the proportion of assertions whose score is at least equal or higher than a minimum score and its score variance is less than a maximum variance. This can be formalized in the following way: let t_i be the i-thassertionand $m_i$be the set of measurements of the i-th triplet, and T the set of all triplets. The criterion *s* is then expressed by the following equation:

$$s = \frac{|\{t_i \; in \; T : var(m_i) < var_{max} and \; mean(m_i) > score_{min}\}|}{|T|}$$

Let var_{max}be 0.4 and $score_{min}$ be 4. Over the whole dataset, this proportion is 0.58. However, assertions from news exhibit a much higher proportion of 0.71, while Wikipedia-sourced assertions show 0.48, a statistically significant difference. This further confirms the difficulty of Wikipedia as a language data source, compared to news articles.

The evaluation of correctness has shown that almost all of the assertions were considered to be correct: only three out of 402 were judged as incorrect. This can be explained by the ambiguity of assertions: since they are not strictly bound to an ontology in all of its parts (only subject or object are optionally assigned an URI), they can be interpreted as correct in some possible domain.

To summarize, the majority of assertions extracted using our methods can be considered informative and therefore sufficiently useful in indexing for information retrieval.

6 Article Template Discovery

The fact mining toolkit also tries to align each document to a topic template. We first define this novel task.

For the purpose of this section, we define a *topic* to be an abstract collection of documents which share some underlying semantic structure. For example, "news articles about bomb attacks" is a viable topic, as is "biographies of physicists".

We can then define a *topic template* as a structure that captures the semantic commonalities of the documents belonging to a topic. For example, in the bomb attack case, we expect many of documents to state that some number of people were killed, that some building was destroyed, that some person performed an attack etc. In the physicist biography case, we similarly expect most of the documents to state that the person was born in some place, studied at some institution, received some award etc. In the simplest case, the template can be simply a bag of such generalized statements.

Finally, we define the task of *aligning* a document to a topic template: given a new document, the goal is to identify the best-matching template for that document and the semantic facts in the document (if any) that are instantiations of (parts of) the template. For example, imagine a document describing the life of Albert Einstein, stating among other thing that "Albert Einstein was born in Ulm". Assuming the system is aware of a "physicist biography" topic and assuming further that the topic's template contains a pattern "person born in place", the desired outcome is to identify the "physicist biography" topic as the correct one and to indicate that the statement "Albert Einstein was born in Ulm" corresponds to the "person born in place" pattern.

6.1 Algorithm

We have developed two pieces of software that together perform the article template alignment task.

The first tool produces a topic template given a set of documents belonging to that topic. We chose a simple representation: in our implementation, each template is a bag of *subject-verb-object* triplets aligned to WordNet, e.g. *person-study-institution*. Note that the topics have to be specified manually and specifying a topic means providing a set of sample documents. We have also considered identifying topics automatically, but early efforts have shown the problem to be hard and the solutions very likely to introduce too much noise into the pipeline.

The second tool takes a single document as input, attempts to align it to all known templates and outputs the best match, if any.

Note that the chosen template representation is different from the one proposed in D2.2.1 – Prototype of the Fact Mining Toolkit[23]. While the prototype followed the same basic steps implied by the task definition, the topic templates were much more structured – the template statements were linked into a graph and there tended to be less redundancy in the statements. Templates produced in this way were more suitable for human inspection, but required a whole *set* of articles about a given news story to have any hope of successful alignment – and even then the recall was extremely low.

6.1.1 Template Construction

This is the phase that constructs a topic template given a set of documents belonging to a topic. In reality, the algorithm requires *two* input sets of documents, the second one being a set of negative examples (i.e. documents not belonging to the topic). In our experiments, this was not an issue as they were based on web documents. Allowing for a low probability of false negatives, we can simply use a set of random documents from the web, possibly written in a similar style.

As our approach is based on the semantic subject-verb-object triplet representation of documents and templates, some **preprocessing** is required. We analyze each input document with Enrycher to obtain subject-verb-object triplets. We map them to WordNet using the "most common sense" heuristic. If no

WordNet concept matches and the triplet constituent of multiple words, we try to map only the head word of the phrase (heuristics for English: the last word for noun phrases, the first word for verb phrases).

In outline, the approach proceeds after the preprocessing as follows:

1. Construct a directed acyclic graph of all corpus triplets and their generalizations (triplets being nodes and hypernymy giving rise to graph edges).
2. Assign a relevance score to each triplet. The scoring function should be constructed so that neither very specific nor very general triplets are scored too high.
3. Cut the graph at some score threshold, i.e. retain only the triplets with score higher than the threshold and their specializations. Further retain only triplets that no longer have a generalization in the cut graph.
4. Specialize the remaining triplets as much as possible while retaining support.

We follow the overview above with some details.

In **step 1**, we count the number of occurrences in the corpus for each triplet. Implicitly, for every triplet (s, v, o) found in the corpus we also log an occurrence of $(s^{\uparrow}, v^{\uparrow}, o^{\uparrow})$ where $s^{\uparrow}$ is either equal to$s$ or its (not necessarily direct) hypernym. This stands to reason: if a document claims *dog-eat-sausage*, then it implicitly also claims *animal-eat-food*. In the graph that we construct from triplets, (s, v, o) is only linked to its direct generalizations (s', v, o), (s, v', o) and $(s, v, o')$where s', v', o' are direct hypernyms.

In **step 2**, we experimented with two different scoring functions. The first one is the classic TF-IDF adapted for triplets:

$$\text{score}_{\text{TFIDF}}(t) = f_T(t) \cdot \log \frac{|C|}{|\{d \in C; t \in d\}|}$$

Here, C is the set of all documents in the corpus and $f_T(t)$ is the number of occurrences of t in the documents for topic T; as discussed in the previous paragraph, the occurrence need not be literal; any specialization of t will do.

The second scoring function is Bayesian in nature: it estimates the probability of a document belonging to topic T given that we observed triplet t:

$$P(T|t) = {P(t|T)P(T)}\Big/{P(t)} = \frac{\frac{f_T(t)+k}{f_T(*)+k\cdot|\{t' \in d \in T\}|}\cdot\frac{f_T(*)}{f_C(*)}}{\frac{f_C(t)+k}{f_C(*)+k\cdot|\{t' \in d \in C\}|}} \propto \frac{f_T(t)+k}{f_C(t)+k}$$

$$\text{score}_{\text{BAYES}}(t) = \frac{f_T(t)+k}{f_C(t)+p\cdot k}$$

Here, $f_T(t)$ is the number of occurrences of triplet t in the topic T; $f_T(*)$ is the number of all triplets in all documents of T. $f_C(t)$and$f_C(t)$ denote the same for the whole corpus. The parameter k is the Laplacian smoothing parameter and was set to 1.

The Bayesian score almost exactly echoes the probability described above; however, we introduce another parameter $p > 0$ to penalize triplets for which the probability estimate is uncertain due to small amounts of data. For example, we prefer a triplet that appears within topic T in 74/100 cases over a triplet that appears within T in 3/4 cases. A way to think of this correction is that we can only get an estimate $\hat{P}$ for $P(T|t)$ from the data; the real, unobservable $P(T|t)$ however is drawn from a probability distribution which peaks at $\hat{P}$. As it turns out that scoring an irrelevant triplet too highly is a much more common problem than the other way around, we do not base our score directly on $\hat{P}$ but rather on a pessimistic estimate of $P(T|t)$ which is expected to have $m\%$ (for some m) of the probability mass to its right. We do not compute m; as p is a function of m and the assumed probability distribution for $P(T|t)$, we rather experimentally directly set $p = 10$.

In **step 3a**, the threshold is determined as the 1000th highest score; we find that this results in patterns that are manageably small (around 100 triplets once all the steps are completed), yet expressive enough. The motivation for **step 3b** is to shed the over-specialized triplets: If the final template contains the triplet *animal-eat-food*, it will implicitly also match documents containing *dog-eat-sausage*, so there is no need to include *dog-eat-sausage*in the template explicitly.

Continuing with the dog example, suppose the target topic T contains *all* the documents with the verb *eat*. Then all the generalizations of *animal-eat-food* will have the same frequencies within the corpus and the topic and thus also the same score. Consequently, step 3 will produce *entity-eat-entity* as a template triplet, which is clearly too general. Analogous cases do indeed occur in real-life data, hence **step 4**. We consider each triplet t from step 3 and replace it with its most frequent specialization $t^{\downarrow}$ if $t^{\downarrow}$ covers at least $\alpha = 80\%$ of the triplets from documents with topic T that t does.

Two minor additional ad-hoc corrections/"hacks" are used in the current implementation, which need to be replaced by a more principled solution in the future. Firstly, the data we used for learning templates was clustered into only about 10 stories for each topic. We thus weighted each article with $\frac{1}{\#\text{ articles in story} + 10}$ to prevent any one story from dominating the topic, causing story-specific triplets to be included in the topic template. The weight was determined by trial and error. Secondly, the scoring functions currently used do not sufficiently penalize triplets that are very frequent both inside and outside the topic, e.g. person-say(_to)-person, causing them to be included in topic templates. We therefore remove from consideration all triplets t for which

$$f_C(t) > 3 \cdot f_T(t)$$

While this solution is in principle sound, the constant 3 is tailored to our specific ratio of topic and corpus sizes and should be generalized.

6.1.2 Document Alignment to Templates

This step is trivial. Each input document is preprocessed as for template construction (i.e. WordNet-aligned triplets are extracted). For each known template, we try to match every input triplet t_{IN} (including the implicit generalizations) with every template triplet t_T. The appropriateness score assigned to the (document, template) pair is the sum of all t_T for which some t_{IN} matches.

If the score is above some manually determined, topic-specific threshold, the article is taken to be a match for the topic. We consider setting the threshold an integral preparing sample documents for a topic (which is a manual process anyway).

6.2 Evaluation and Illustrative Examples

We evaluated our solution on three different topics:

- "bomb" – News articles on bomb attacks
- "layoffs" – News articles on companies laying off workers
- "visit" – News articles on politicians' official visits

The data for all three topics comes in the form of internet news articles. We used the web crawler described in deliverable D1.3.2 [25]to obtain the articles and manual keyword-based search to identify sample documents for the topics. All the documents used in the experiments are by nature grouped by the news story they cover; typically, about 50 articles cover a story. The grouping is used neither in the algorithm nor during evaluation; it is however good to be aware of the hidden additional structure it provides.

As the quality of a template in its own right is hard to evaluate numerically, we present the top 10 template triplets for each of the topics according to the TF-IDF metric (which does slightly better for our taste). A complete template contains about 100 triplets regardless of the topic.

Example 2. Top 10 template triplets for each of the topics according to the TF-IDF metric.

Topic "bomb"
- bomb -- blow -- himself*
- person/individual -- kill -- attack/onslaught
- bomber -- blow -- himself*
- group/grouping -- claim -- duty/responsibility
- he* -- blow -- himself*
- bomb -- kill -- people
- one/1 -- claim -- duty/responsibility
- civilian -- die/decease -- attack/onslaught
- attack/onslaught -- come/come_up -- day/twenty-four_hours
- bomber -- explode/detonate -- explosive

Topic "layoffs"
- it* -- cut -- occupation/business
- environment -- be -- challeng*
- we* -- adapt/accommodate -- environment
- we* -- be -- abl*
- it* -- extinguish/eliminate -- occupation/business
- cut -- be -- part/portion
- environment -- stay/remain -- time/clip
- we* -- necessitate/ask -- people
- company -- have/have_got -- employee
- one-half/half -- descend/fall -- history

Topic "visit"
- Asian_country/Asian_nation -- be -- member/fellow_member
- ECO* -- unite/unify -- Asian_country/Asian_nation
- person/individual -- meet/run_into -- person/individual
- sebaceous_cyst/pilar_cyst -- visit/see -- country/state
- person/individual -- attend/go_to -- teaching/instruction
- rebellion -- stage -- person/individual
- person/individual -- give -- person/individual
- person/individual -- travel/go -- country/state
- organization/organisation -- establish/set_up -- Asian_country/Asian_nation
- BSEC_member_st* -- be -- country/state
- person/individual -- attend/go_to -- summit/height

Note: concepts marked with an asterisk (*) are not present in WordNet and are represented by the stem of their originating word(s).

Although some of the triplets are obfuscated by their (possibly erroneous) conversion to a WordNet and back to (possibly context-inappropriate) text, we can see that most of them are indeed relevant to the topic. Some triplets are clearly generalized and will form template slots (e.g. *person-travel-country*) while others are quite fixed (e.g. *environment-be-challenging*). We believe that the latter belong in a template as well in that they are "what is typically told in documents on this topic" and in that they provide context for the remaining triplets. We therefore intentionally do not discriminate against triplets that do not have diversified instantiations in documents.

To obtain a numeric evaluation result, we turn to indirect evaluation. We measure for each of the topics how good the template is at classifying new, unseen articles as belonging to the template's topic or not. This is the training/evaluation split we used:

Table 2.Dataset sizes and splits.

	Training		Evaluation	
	articles	**stories**	**articles**	**stories**
bomb	580	9	364	3
layoffs	741	17	260	5
visit	391	7	103	2

The table above only represents the positive examples, i.e. documents belonging to a topic. In addition, we included – to both the training and the test set – about 4000 random news articles from a large database which are with high probability negative examples for any topic. When performing the data split, we were careful to put all articles covering a single story in the same set. This prevents story-specific facts from having too much influence and ensures we measure only the effect of topic-specific features/triplets.

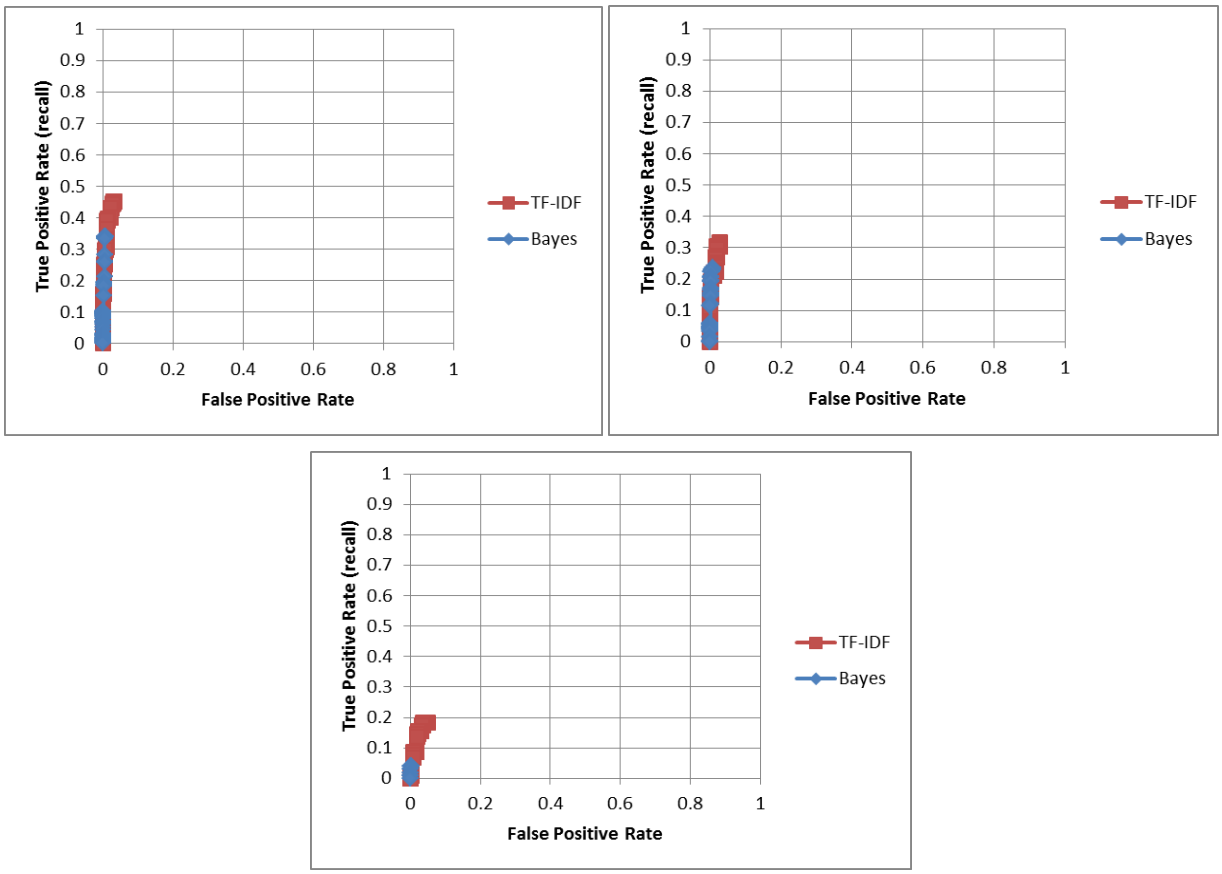

Figure 12. The ROC curves for the classification task on the a) bomb b) layoffs and c) visit datasets. Note the inherently low recall and high precision and the highly topic-specific performance.

Figure 12 gives the ROC curves for the topic classification task as an indirect measure of template's quality. Notice the unusual shape: even the smallest possible increase in the cutoff value (= score value separating positively and negatively classified examples) results in a sharp drop in recall and a sharp rise in precision. While the second is welcome, the first is not; we identified several probable possible sources of error:

- Data sparsity. Although triplets to some degree normalize the way we express facts, a semantic fact in the true sense of the word can often still be expressed with many different triplets. For the template-relevant (true) facts, some of their triplet expressions never appear in the training data but do so in the evaluation data. A larger training set would help here.

- Triplet extraction. Triplet extraction is a very hard and not properly solved problem in itself; consequently, what our algorithm regards as input data is already quite rich in noise, i.e. incorrectly extracted triplets. In addition, extractors suffer from relatively low recall, i.e. they fail to extract triplets present (although obliquely) in the text. A better triplet extractor would help here.
- Template construction and size. Even when triplets are extracted successfully and correctly, they might not get included in the template because the scoring function is faulty or the template is inherently too small (there are more useful facts to be included than the currently allowed number of triplets in a template). A better scoring function would help, perhaps one based on some manually created training data (which triplets are template-worthy?). Bigger templates might help as well, but with the obvious drawback of decreased template clarity.

Additional analyses would be required to identify where there is the most room for improvement.

Some more conclusions can be drawn from the data above:

- Performance is highly dependent on the topic. Having a very specific topic helps considerably. However, it is hard to measure how "well" the algorithm does for a specific topic: the subjectively judged quality of the template itself and template's performance in the classification task do not correlate much. Compare the "layoffs" and "visit" topics – the first produces a poor template that however does well in classification, possibly because its triplets are not very general.
- The Bayes-based templates perform with slightly higher precision than the TF-IDF ones for a given recall level. However, the maximal recall attainable to them is lower.

Returning to qualitative analysis, here are some examples of how documents were aligned to the template. On the left, we have the triplet extracted from the input document. On the right, we have its generalization which also appears in the template.

Example 3. The way documents were aligned to the template.

Topic "bomb"

policeman/police_officer -- hit -- body/organic_structure -> person/individual -- move/displace -- body/organic_structure

group/grouping -- claim -- duty/responsibility -> group/grouping -- claim -- duty/responsibility

attacker/aggressor -- open/open_up -- fire -> person/individual -- open/open_up – fire

Topic "layoffs"

Sullivan/Louis_Sullivan -- state/say -- statement -> person/individual -- state/say -- statement

share/portion -- rise/lift -- percentage/percent -> assets -- travel/go -- percentage/percent

president -- state/say -- statement -> person/individual -- state/say -- statement

Topic "layoffs"

it* -- cut -- occupation/business -> it* -- cut -- occupation/business

it* -- shutter -- factory/mill -> it* -- close/shut -- factory/mill

sale -- drop -- percentage/percent -> sale -- move/displace -- percentage/percent

Topic "visit"

president -- give -- Elizabeth/Elizabeth_II -> person/individual -- give -- person/individual

president -- give -- queen -> person/individual -- give – queen

she* -- volunteer -- plan/program -> person/individual -- communicate/intercommunicate -- plan/program

she* -- volunteer -- plan/program -> person/individual -- communicate/intercommunicate -- idea/thought

While each of the above examples showcases some imperfections – the quality is clearly not the same as it would be if the task was performed manually – the results are overall sensible, understandable and relevant.

6.3 Usage

The real-time article template discovery tool is included in the fact mining toolkit as part of the Enrycher service-oriented architecture: if an input article matches one of the known topic templates, the matching statements from the document are output along with their corresponding template entries.

The template construction part remains a separate piece of software. The project partners are welcome to suggest new topics to Mitja Trampus (JSI); we will build the templates and integrate them into Enrycher.

7 Conclusions and Future Work

In this deliverable we described the general RENDER architecture in Section 2, mainly focusing on the Fact Mining Toolkit. In Section 3 we discussed the integration of the Fact Mining Toolkit with the other components of the RENDER architecture, especially with the Knowledge Diversity Ontology and ontologies from the Reference Knowledge Stack (mainly PROTON) which are used to represent the extracted textual information in RDF.

Section 4 described the knowledge infrastructure and faceted search, focusing on the data layer infrastructure and KIM faceted search.

Furthermore, we evaluated the Fact Mining Toolkit in terms of the informativeness and correctness of the extracted facts (see Section 5). Our findings show that the majority of assertions extracted using our methods can be considered informative and therefore sufficiently useful in indexing for information retrieval.

In Section 7 of this deliverable we presented the article template discovery toolkit component.

In the following sub-section we discuss more in-depth conclusions and future work directions related to the knowledge infrastructure and faceted search, as well as the article template discovery approach.

7.1 Knowledge Infrastructure and Faceted Search

The next stages of this work will be in integrating information about opinions and sentiments into the data layer infrastructure, extending the options of the faceted search to cover the entire RKS, and analyzing the effects of the integration of the data produced by Enrycher fact mining tool into the data layer infrastructure, e.g. FactForge, with respect to the quality of the data and the wealth of information available for querying. Additionally, we will experiment with the data when the fact mining pipeline described in section will be fully integrated with OWLIM [13], and work on integrating CLAS toolkit (see RENDER technical architecture above) and analyzing its effects on the fact mining.

7.2 Article Template Discovery

We believe the template construction and alignment task to be a novel and interesting problem with useful real-life applications, meriting future improvements. In particular, we plan on further exploring the following areas:

Sparsity. While WordNet offers data representation that is less sparse than bag of words, we would still need impractically large quantities of training data to obtain reliable statistics for all possible triplets. Additionally, the WordNet graph is not always as dense as it could be; for example, there is no notion of related concepts. As a consequence, there is little transfer of relevancy information between related triplets. We plan to define a relatedness measure between triplets (based on their constituents), hoping to facilitate such information transfer and thus obtaining better statistics with less input data.

Reintroducing template structure. Deliverable D2.2.1 Prototype of the Fact Mining Toolkit envisioned topic templates in the form of small semantic graphs. The original approach was later abandoned because the resulting templates were too rigid to match single documents (which contain a relatively low number of extracted triplets). However, a more structured template is more informative to the human observer as it provides additional context for some of the nodes. We will therefore attempt to maintain the current approach but extend it by constructing a graph (subjects and objects being nodes, verbs being relations) from the bag of triplets that is now a template – but this graph will only be used for the presentation of results, not constricting the (already strict) matching criteria.

We plan to tackle this task with the help of collections of articles covering a single story: although they might report on the same (or related) facts using different triplets, entities (e.g. "Germany", "Obama",

"police station") should remain largely fixed across the articles. Hopefully, we can exploit this to identify slots (i.e. the generalized entities in triplets) in the template that should be unified (i.e. represented as a single node in the final template graph).

References

[1] Damova, M., Kiryakov, A., Grinberg, M., Bergman, M.K., Giasson, F., Simov, K. *Creation and Integration of Reference Ontologies for Efficient LOD Management*. In: Semi-Automatic Ontology Development: Processes and Resources, IGI Global, Hershey PA, USA, Dr. Armando Stellato and Dr. Maria Teresa Pazienza (Eds.), February 2012.

[2] *DMOZ.* (2012). Retrieved from http://www.dmoz.org.

[3] *DBpedia.* (2011). Retrieved from Structured information from Wikipedia: http://DBpedia.org.

[4] *Dublin Core.*Retrived from http://dublincore.org/documents/dces/.

[5] *Faceted Search*. Retrived from Wikipedia. http://en.wikipedia.org/wiki/Faceted_search.

[6] *FactForge*. (2012). Retrieved from A Reason-able View to the Web of Data: http://factforge.net, http://www.ontotext.com/factforge.

[7] *Freebase*. (2012). Retrieved from http://www.freebase.com.

[8] *Geonames*. (2011). Retrieved from A geographical database: http://www.geonames.org.

[9] *HEO* (Human Emotions Ontology). Retrieved from

[10] Kiryakov, A., & Momtchev, V.*Two Reason-able Views to the Web of Linked Data*. Paper presented at the Semantic Technology Conference . San Jose, USA, June 2009.

[11] Kiryakov, A., Tashev, Z., Ognyanoff, D., Velkov, R., Momtchev, V., Balev, B., & Peikov, I. *D5.5.2. Validation goals and metrics for the LarKC platform.* FP7 – 215535 LarKC project deliverable, 2009.

[12] *KDO* (Knowledge Diversity Ontology). Retrieved from http://labs.mondeca.com/dataset/lov/details/vocabulary_kdo.html.

[13] *OWLIM.* (2012). Retrieved from http://www.ontotext.com/owlim.

[14] *PROTON*. (2012). Retrieved from http://www.ontotext.com/proton.

[15] *PROTON documentation*. (2012). Retrieved from http://www.ontotext.com/proton General documentation.

[16] *SIOC* (Semantically-Interlinked Online Communities)Retrieved fromhttp://sioc-project.org/.

[17] *WordNet*. (2011). Retrieved from A lexical database for English: http://wordnet.princeton.edu.

[18] *YAGO* (2012). Retrieved from A Spatially and Temporally Enhanced Knowledge Base from Wikipedia: http://www.mpi-inf.mpg.de/yago-naga/yago/

[19] *OpenCyc* (2012). Retrieved from OpenCyc for the Semantic Web: http://sw.opencyc.org/

[20] *Telefonica Taxonomy* (2012). Retrieved from: http://topics.render-project.eu/telefonica

[21] *Crowdflower*(2012). Retrieved from*:* http://crowdflower.com/

[22] *Enrycher* (2012). Retrieved from http://enrycher.ijs.si/

[23] B. Fortuna, D. Rusu, M. Trampus, L. Dali, T. Stajner, M. Grobelnik. *Prototype of the Fact Mining Toolkit*. RENDER Project Deliverable D2.2.1. 2011.

[24] M. Grinberg, M. Damova, A. Kiryakov. *Initial data integration*. RENDER Project Deliverable D1.2.1. 2011.

[25] D. Vrandecic, M. Trampus, B. Novak. *Initial corpora collection (English, French, German, Italian, Spanish)*. RENDER Project Deliverable D1.3.2. 2012.

[26] A. Thalhammer, I. Toma, R. Hasan, E. Simperl, and D. Vrandecic, *How to Represent Knowledge Diversity*, Poster, ISWC 2011.

[27] J.G. Breslin, A. Harth, U. Bojars, S. Decker, *Towards Semantically-Interlinked Online Communities*, Proceedings of the 2nd European Semantic Web Conference (ESWC '05), LNCS vol. 3532, pp. 500-514, Heraklion, Greece, 2005

[28] M. Damova, A. Kiryakov, K. Simov, S. Petrov. *Mapping the central LOD ontologies to PROTON upper-level ontology*. Ontology Mapping Workshop at ISWC 2010, Shanghai, China, November 2010.

[29] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan. *GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications*. Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, July 2002.

[30] R. Likert. *A Technique for the Measurement of Attitudes*. Archives of Psychology 140: pp. 1–55, 1932.

