

Document identifier: Decipher-D4.2.1-WP4-BUT Relationship mining component-PU

Project title: Digital Environment for Cultural Interfaces;  
Promoting Heritage, Education and Research

Project acronym: DECIPHER

Project identifier: FP7-270001-Decipher

Partners: Dublin Institute of Technology  
National Gallery of Ireland  
Irish Museum of Modern Art  
Open University  
System Simulation Limited  
Brno University of Technology  
Alinari 24 ORE SpA

Author(s): Pavel Smrz, Lubomir Otrusina, Jan Kouril, and  
Jaroslav Dytrych  
Brno University of Technology

Version: v01

Type: Report (Deliverable D4.2.1)

Report availability: Public

Date: July 31, 2012



## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Component architecture</b>	<b>5</b>
2.1	NL pre-processor . . . . .	5
2.2	On-demand NL processor . . . . .	7
2.3	Content classifier and analyser . . . . .	8
2.4	Information extractor . . . . .	11
2.5	Annotation server . . . . .	12
2.6	SEC Store API . . . . .	13
<b>3</b>	<b>Methods</b>	<b>14</b>
3.1	Statistical classification and feature representation . . . . .	14
3.2	Measures of semantic relatedness . . . . .	17
3.3	Finite state machines and extraction patterns . . . . .	18
<b>4</b>	<b>Evaluation</b>	<b>22</b>
4.1	Relevance classifier . . . . .	22
4.2	Content acceptability estimator . . . . .	24
4.3	Theme classifier . . . . .	26
4.4	Opinion miner . . . . .	30
4.5	Relation extractor . . . . .	31
<b>5</b>	<b>Integration with other project components</b>	<b>35</b>
5.1	Annotation use case . . . . .	35
5.2	Search semantically enriched documents . . . . .	36
<b>6</b>	<b>Conclusions and future directions</b>	<b>38</b>

## 1 Introduction

The term *relationship*, referring to the way in which two things are connected, is understood in a broad sense in this deliverable. In addition to relations among named entities in a particular piece of text, it involves relations between a topic and a document, an intended audience and a text, semantic similarity (relatedness) between words or phrases, etc. The Semantic Enrichment Component (SEC) developed within the DECIPHER project provides key functionality for automatic identification of all these kinds of relations. Respective methods are described in this deliverable.

*Text annotation* is a unifying concept that joins results of individual relation identification procedures (both – manual and automatic). It also provides a natural way to integrate them into a user interface. Users of the DECIPHER system can annotate texts (i. e., provide metadata) to express relations. For example, they can tag documents as relevant to a particular event, mark conflicting views on an exhibition, or identify influences of a painter expressed in a text. The StorySpace takes into account the annotations and employs the metadata in standard and exploratory searches, when suggesting related heritage objects, visualizing relations, etc. Figure 1 shows an example of a visualisation resulting from the following annotations:

```
<influencee>Jack B. Yeats</influencee>  
<rel_realization>was influenced</rel_realization>  
in his early days by the work of  
<influencer>Goya</influencer>.
```

```
<painter>Jack B. Yeats</painter>  
<rel_realization>was brother of</rel_realization>  
the Nobel Prize winning poet  
<author>William Butler Yeats</author>.
```

(Note that the annotations are simplified to demonstrate only the roles and relations shown in the visualization.)

An analysis of user requirements showed that museum professionals prefer a full control of information that is entered, stored and managed by the StorySpace. At the same time, manual annotation of texts presents a tedious task. That is why the concept of annotation suggestions was introduced – the DECIPHER system identifies potential relations (relation mentions) in a text and users can just accept or reject the suggestions.

This deliverable evaluates results of automatic methods that generate annotation suggestions in various DECIPHER tasks. Obviously, user satisfaction depends on the quality of these processes – if the SEC proposes irrelevant documents as potentially related, incorrectly identifies events that have nothing to do with a heritage object, or it makes other errors, users

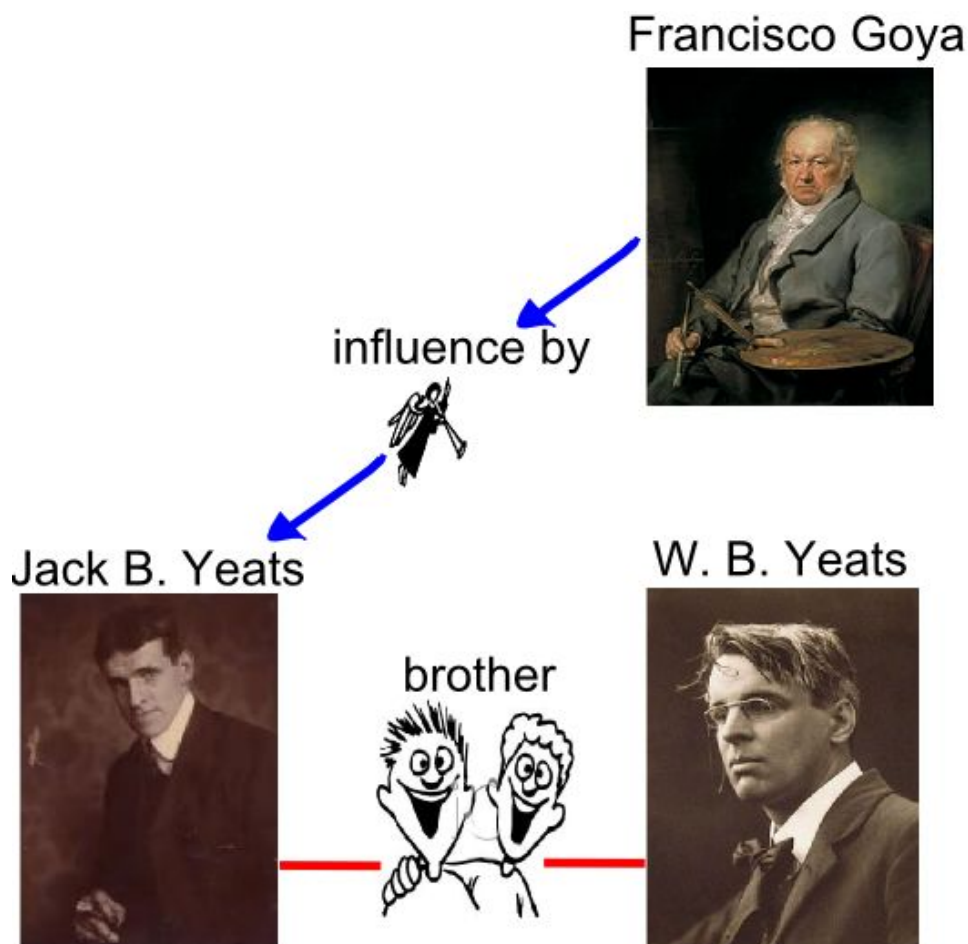


Figure 1: An example of visualization of influence relations

will spend more time in declining suggestions or modifying proposed annotation structures. It is therefore crucial to search for robust annotation methods that really facilitate the use of the DECIPHER system.

The rest of the deliverable is organized as follows. The following section describes individual modules (software packages) of the SEC and discusses how they are interconnected. Section 3 presents implemented methods that are incorporated in the current version of the SEC. Section 4 focuses on experiments and results on datasets collected within the project. Section 5 details integration of the SEC with other key components – the Aggregator and the StorySpace. We conclude with directions of a future work.

## 2 Component architecture

The Semantic Enrichment Component (SEC) processes semistructured or unstructured text employing clustering, classification, and information extraction techniques to produce a variant of the text with semantically annotated entities and relations. Outputs can be transformed to an RDF describing events and other entities according to the CIDOC CRM. The SEC also supports an interactive service for annotating tracts of texts. Last but not least, it estimates reliability of extracted semantic knowledge.

The SEC can be invoked by the DECIPHER Content Aggregator. This would typically correspond to a situation in which the Focused crawler acquired a set of texts and the Aggregator would schedule its processing by the SEC. The Similarity Recommender UI subcomponent of the StorySpace depends on the SEC functionality as well. It communicates with the Similarity engine subpackage and receives suggestions on potentially related entities to be presented to a user.

The Annotation editor – a subpackage of the StorySpace Resource UI – accesses the Annotation server and makes accessible a user feedback provided in the process of assisted annotation. A specific set of annotations is finally transformed to StorySpace internal structures to enable immediate visualization.

The SEC consists from six major components:

1. Natural Language (NL) pre-processor
2. On-demand NL processor
3. Content classifier and analyser
4. Information extractor
5. Annotation server
6. SEC Store API

Figure 2 shows the SEC decomposition schema and interconnected packages from the Content aggregator and the Story-Space.

The subcomponents communicate internally to achieve their respective goals. For example, the Information extractor can ask for specific processing of a text (such as parsing or co-reference resolution) based on a request for event template slot filling sent to it by the Annotation server.

### 2.1 NL pre-processor

This subcomponent comprises basic pre-processing steps that are useful for other tasks dealt with within the SEC. As opposed to the next – the On-demand NL processing (NLP) subcomponent, the low-level analysis is

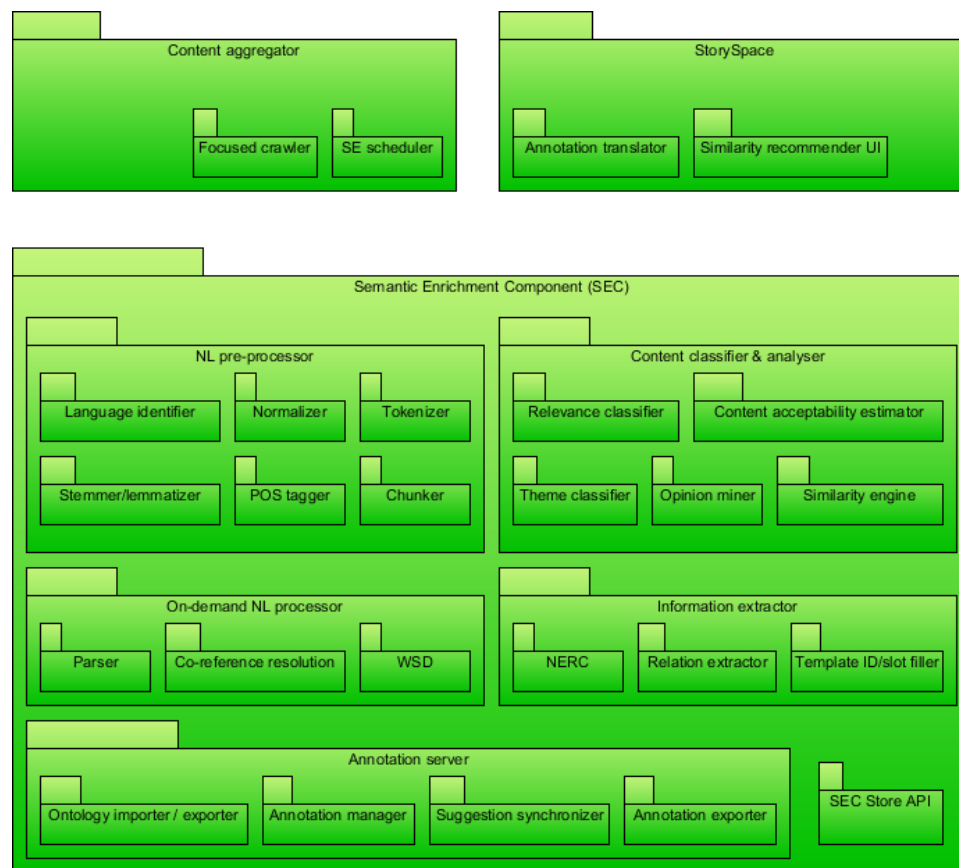


Figure 2: Architecture of the Semantic Enrichment Component

supposed to be available for all languages the DECIPHER system would be deployed for (currently English and Italian).

As a part of the content pre-processing step, the SEC recognizes the language (and encoding) in which an input text is written. An open source tool – Chromium Compact Language Detector<sup>1</sup> – is employed for this task. Only documents written in a particular language of interest are kept in the pipeline, the rest is filtered.

Depending on the recognized language, text normalizer then harmonizes spelling and capitalization and cleans up unnecessary metadata. This is followed by tokenization which breaks up the stream of characters into distinct meaningful units – tokens. Tools available in the NLTK<sup>2</sup> (Natural Language Toolkit) are used for the above-mentioned tasks.

<sup>1</sup><http://code.google.com/p/chromium-compact-language-detector/>

<sup>2</sup><http://nltk.org/>

Stemming and lemmatization aim at reducing inflectional forms and some derivationally related forms of a word to a common base. English and Italian Snowball<sup>3</sup> stemmers are used in the current system. Furthermore, Morph-It!<sup>4</sup> is applied on the Italian data (on the TreeTagger output – see below).

TreeTagger<sup>5</sup> with English and Italian parameter files are responsible for PoS (Part of Speech) tagging (assigning grammatical categories to words in a text depending on their contextual use). TreeTagger can also be used as a chunker (a tool identifying noun and verb phrases) for English. Only basic chunks are identified for Italian by means of a set of regular expressions run on a TreeTagger output.

## 2.2 On-demand NL processor

Full linguistic processing of large amounts of text available to the DECIPHER system (e. g., cached by the Focused crawler package) can take a lot of time. Moreover, expensive processing is not always necessary – either tools further in the pipeline provide sufficient results on a pre-processed text only, or the quality of generic NLP tools employed in this subcomponent is not sufficient (for a given language) and it is beneficial to deal with a specific task later, when a context is already known. That is why packages briefly described in this subsection are not run on all the input, but rather wait for an explicit invocation by a component that needs particular results of the NLP package.

Various parsers are available for English. Unfortunately, their results, time efficiency and the way they encode their output differ significantly. After several comparisons on data relevant to DECIPHER, three dependency parsers were chosen. MiniPar<sup>6</sup> provided the best trade-off between accuracy and time/memory demands. DeSR<sup>7</sup> is also fast and is available for both – English and Italian. Stanford Parser<sup>8</sup> is relatively slow but generates a base for the Stanford Deterministic Coreference Resolution System.

Trained parsing models are not generally available for Italian; DeSR is an exception. According to a recent comparison, the best Italian dependency parser is ParsIt<sup>9</sup>. Unfortunately, it is not freely available. Another general system – TextPro<sup>10</sup> is freely available for research purposes only.

<sup>3</sup><http://snowball.tartarus.org/>

<sup>4</sup><http://dev.sslmit.unibo.it/linguistics/morph-it.php>

<sup>5</sup><http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>

<sup>6</sup><http://webdocs.cs.ualberta.ca/~lindek/minipar.htm/>

<sup>7</sup><https://sites.google.com/site/desrparser/>

<sup>8</sup><http://nlp.stanford.edu/software/lex-parser.shtml>

<sup>9</sup><http://www.parsit.it>

<sup>10</sup><http://textpro.fbk.eu/>

Co-reference resolution in English is performed by the above-mentioned Stanford Deterministic Coreference Resolution System<sup>11</sup>. For compatibility reasons across languages, we have also experimented with BART<sup>12</sup> which was recently trained for Italian as well. Unfortunately, both the tools provided unsatisfactory results on the DECIPHER dataset so that a domain-specific co-reference resolution engine seems to be unavoidable.

WSD (Word Sense Disambiguation) tries to identify which meaning of a polysemous or homonymous word applies in a particular context. Poor performance on the collected data characterizes WSD tools we experimented with. Neither GWSD (unsupervised Graph-based Word Sense Disambiguation system)<sup>13</sup>, LingPipe WSD modules<sup>14</sup>, Duluth Senseval-2 scripts<sup>15</sup>, UKB<sup>16</sup>, nor other available experimental tools provided sufficient accuracy. Limited training data calls probably again for an application-specific solution which will take into account results of further processing, especially outputs of the Named entity recognizer and classifier. This forms a direction of our future work.

### 2.3 Content classifier and analyser

This subcomponent deals with classification of textual content w.r.t. general relevance, theme, and appropriateness for specific user groups (such as children visitors). It also analyses opinionated texts (e.g., social media reflections of specific exhibitions) and provides key functionality for similarity-based recommendations. Major subpackages correspond to the work planned in Task 4.2.

Relevance classifier is used to decide whether a document (e.g., downloaded from the web by the Focused crawler) is generally relevant for the DECIPHER domain (dealing mainly with visual art) and/or whether it corresponds to one of predefined categories (e.g., whether it is a biography). For example, the classifier is able to distinguish that a document *Paul Henry discovered on Antiques Roadshow* is relevant for a query on the Irish painter, while *Paul Henry slammed for asylum seeker comments* is not (it rather refers to a controversial broadcaster). The classifier is based on supervised machine learning. An implementation of the SVM (Support Vector Machine) model as provided by the libsvm library<sup>17</sup> is employed. The training dataset corresponds to the data on Visual Art from Freebase and a subset of Wikipedia

<sup>11</sup><http://nlp.stanford.edu/software/dcoref.shtml>

<sup>12</sup><http://www.bart-coref.org/>

<sup>13</sup><http://www.cse.unt.edu/~rada/downloads.html#gwsd>

<sup>14</sup><http://alias-i.com/lingpipe/demos/tutorial/wordSense/read-me.html>

<sup>15</sup><http://www.d.umn.edu/~tpederse/senseval2.html>

<sup>16</sup><http://ixa2.si.ehu.es/ukb/>

<sup>17</sup><http://www.csie.ntu.edu.tw/~cjlin/libsvm/>



pages related to the same subject. Negative examples were taken from other, non-relevant Freebase categories and Wikipedia pages linked from them.

As the classifier takes advantage of the machine learning approach, specific document categories to be distinguished can be defined as a part of the SEC administration. Of course, a relevant training set needs to be provided. The current version of the Relevance classifier embodies only two specific document categories – documents containing mainly biographical information (a primary source for the event extraction task) and documents dealing with the art market (auctions, prices, etc.). The former was trained on artist biographies from Wikipedia as well as data collected from specific sites such as <http://www.biography.com/> or <http://www.artist-biography.info/>, the latter on texts from web sites owned by large auction houses and services such as <http://www.artprice.com>.

The Content acceptability estimator involves categorization of reading levels (distinguishing content appropriate for primary schools, secondary schools and adults), characterizing interestingness of texts (the power of attracting or holding attention of a particular user group), and “R-Rating” (predicting content suitability for specific age groups). Machine-learning methods (SVM a CRF – Conditional Random Fields) were employed.

Specific features characterizing the form rather than the content were introduced for the Reading level identifier. They take into account lengths of sentences, relative proportion of nouns, adjectives and verbs, etc. Results can be used for filtering purposes, especially in context-dependent searches.

The component characterizing interestingness of a text focuses on short documents, e. g., descriptions of individual stories. This component aims at identifying that stories such as *Gabriel Metsu had a conflict with his neighbour, Abigaël Ides, who had stolen one of his hens* can bring some colour to the biography of the painter and make it more vivid. The module takes into account the proportion of verbs and other features mentioned above, but also specific lists of terms that explicitly suggest potential interestingness.

Unfortunately, vividness of stories often comes with sexual-sensitive content that can be inappropriate for children (e. g., the above story continues *Having being confronted with the theft, Abigaël accused the artist of being “a whore-hopper”*). That is why the classifier estimates interestingness together with a text “R-Rate”. Filtering then can take into account both the components. The same machine learning approach as above is applied in this classifier but the feature vector extraction takes advantage of other lists of terms – potentially sensitive words.

As discussed in Deliverable D3.4.1, the theme property linked to a DECIPHER dossier provides basic means for organizing its content. The Theme classifier package is a supervised machine-learning module that works with predefined theme categories and classifies documents along these lines. An

SVM model is employed again. To demonstrate implemented functionality, the current version identifies documents describing artworks related to specific categories such as religion or myth themes (containing religious or mythological symbols or conveying religious or mythological ideas). The SEC administrator can add new themes and relevant datasets. In general, the module can be employed to distinguish even fine-grained subthemes, corresponding, e.g., to the “art genes” suggested by the new Art.sy service<sup>18</sup>.

The Opinion miner package analyses opinionated text fragments. Its primary aim is to characterize sentiment (positive/neutral/negative) of blog posts and other user-generated texts related to a specific subject (e.g., reporting on a museum exhibition). As different views on a same topic can arouse reader’s interest, the module also deals with identification of controversies and extraction of opinions on opposite sides. For example, it can identify that the text: *Gabriel Metsu is best known today as a lesser Vermeer. Vermeer’s accomplishment seems a little narrow. Metsu was certainly more prolific than Vermeer, and probably more versatile.* compares the two painters and that it presents an opposite view when compared to: *Metsu is not a patch on the master. Looking at a Metsu after a Vermeer is like reading Fanny Burney after Jane Austen: entertaining in its way but a stern reminder of why the other is so much admired.* In addition to a general sentiment analyser trained on manually annotated data, the package uses specific lists of terms expressing polemic statements and extraction patterns for particular situations, such as the comparison of two artworks/artists.

The Similarity engine offers a set of methods implementing various term and phrase semantic similarity and relatedness measures. First, it takes advantage of manually created resources such as the Wordnet, domain vocabularies and thesauri. Lists of near-synonyms are also populated from Wikipedia alternative names and redirections. Using all these resources, the DECIPHER system can correctly identify that, e.g., a Wikipedia page describing the biblical story on the Binding of Isaac<sup>19</sup> is related to the Gabriel Metsu’s painting *The Sacrifice of Isaac* (even though it is not mentioned on the page) and interlink the two resources.

The second set of semantic similarity methods covers statistical similarity. Large collections of general as well as domain-specific documents needed to be processed. The English Gigaword<sup>20</sup> and the Italian itWaC<sup>21</sup> textual corpora were used for general-language term similarity computations. Domain-specific models were computed from data collected within the DECIPHER project.

<sup>18</sup><http://art.sy/>

<sup>19</sup>[http://en.wikipedia.org/wiki/Binding\\_of\\_Isaac](http://en.wikipedia.org/wiki/Binding_of_Isaac)

<sup>20</sup><http://www ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2003T05>

<sup>21</sup><http://wacky.sslmit.unibo.it>

Advanced latent variable techniques (pLSA – Probabilistic Latent Semantic Analysis, LDA – Latent Dirichlet Allocation and ESA – Explicit Semantic Analysis) were also applied to generalize contexts taken into account during the statistical similarity computation. Resulting distance or similarity measures are stored as separate models that can be explicitly accessed by the Similarity-based recommender and other DECIPHER modules.

## 2.4 Information extractor

Extraction of information on entities and their relations from a text forms a crucial part of the SEC. This is also reflected by the duration of corresponding Task 4.3 which started in M13 and will last till M33. Design and implementation of individual packages correspond to the state-of-the-art methods in the field that were described in deliverable D4.1.1.

All packages described below take a (pre-annotated) text as an input, identify entities, relations or templates and attributes, and enrich the annotation by newly added knowledge. Results can be stored as such or they can be transformed to a set of RDF triples corresponding to an underlying ontology.

Methods that combine functionality of two components depicted as individual packages, e. g., a joint classifier of named entities and relations (see D4.1.1 for a discussion on advantages of such approaches), are subsumed into the module further in the abstract pipeline (the Relation extractor in the example).

The NERC – Named Entity Recognizer and Classifier – takes care of initial identification of relevant entities: names of people, places, artworks, etc. Matching of candidate terms (that correspond to an entity) is based on a FSA (Finite State Automaton) technology. Data for each particular category (e. g., a list of geographical names) is stored in a separate file. A joint list is compiled into a FSA that includes all terms together with their subcategorization.

A disambiguation phase takes into account a context in which a particular NE candidate string appeared and decides whether it really corresponds to an entity in question. In case of ambiguity, it also distinguishes which entity is actually referred to in the text. An SVM-based classifier is employed for this task. If there is a document linked to an ambiguous entity (e. g., a Wikipedia page on a given person), it is used to train the classifier. If no documents are linked to an entity on an ambiguity list, the word expressing a category of the entity or any other available context is used, expanded by the Similarity engine and searched for by the Focused crawler. Pseudo-user feedback is then employed to train the entity disambiguation model.

The Relation extractor identifies textual fragments that express a particular relation between two or more entities (or values). It can work on a

simple PoS-tagged data but it can also ask the Parser package to analyse a particular piece of text. Relation triggers – specific expressions signalling that a relation holds, most frequently relation-denoting verbs – form key elements of the identification process. A CRF classifier is further employed to mark beginnings and ends of particular elements appearing in a relation.

To demonstrate functionality of the package, the DECIPHER system contains a trained model that identifies plot relations of various influence types (see deliverable D3.4.1 for their delineation). It was trained on data collected within the project as well as on a subset of the Freebase dataset expressing influence relations. Resulting annotations are used by the Event recommender.

The template identification and slot filling component corresponds to the task of specific event extraction. From a technical point of view, events connect more relations to a joint structure. In the DECIPHER project, a pre-defined set of event types with corresponding event templates is managed by the SEC administrator. If a new event type is added, relevant data needs to be collected and a specific model trained.

## 2.5 Annotation server

The Annotation server provides a backend for the annotation functionality – it stores types of annotations (which correspond to annotation templates anchored in the DECIPHER ontology, instantiating the general CIDOC/CRM schema), annotated versions of texts, etc. The annotation editor interacts with the server using a specific protocol.

The Annotation server also converts confirmed annotations to a specified format and sends it to the Annotation translator in the StorySpace. This link supports live changes in visualisations (e. g., in a timeline if a user confirmed an annotation of a new relevant event).

The Ontology importer and exporter package can read any ontology in the OWL format and instantiate internal annotation template structures. An administrator can refine and extend the templates to reflect specific annotation tasks (e. g., limit a domain by a list of values). Depending on an administrative setting of the Annotation server, users can also suggest changes in the templates, e. g., add a property to an event template. If accepted by an administrator, the changes are reflected in internal structures and can be exported in the OWL format again.

The Annotation manager forms a core of the Annotation server. It manages internal knowledge structures and all the work on annotated textual fragments. Among others, it localizes the fragments in documents. A location is given by a path in the document object model (DOM), an offset and a size. The representation is robust to changes in general formatting. Moreover, it is usually not necessary to process the whole document to find an annotated textual fragment. For example, a web page boilerplate and other

parts that are not in a DOM node on the path to an annotated fragment will be ignored.

The Annotation manager also handles document internal URIs. The URI of an annotated document identifies a copy of the document that is stored on the server. The annotation process starts with a synchronization step in which a client sends a document URI and its content to the server. The server returns a URI of a local copy of the document which will be used in annotations. This procedure enables annotating documents that the server could not access directly.

The Suggestion synchronizer package realizes communication with all instances of the Annotation editor. A general annotation exchange protocol is used for this purpose. Suggestions are identified by an annotation ID (URI), type, time of creation, its author, URI of an annotated document (or its server copy), XPath to an annotated textual fragment, its offset, length and textual content, annotation content and a specification of annotation attributes. A corresponding RDF Schema is available at <http://nlp.fit.vutbr.cz/annotations/rdfs/annotation-ns.rdf>.

The protocol enables two-way asynchronous communication between clients and servers. If a user adds an annotation, the server sends it immediately to all other users that annotate the same document and are subscribed to a given channel (defined by an author, a group or an annotation type). Changes of annotation types, of the document content and of relevant settings are distributed immediately as well.

The Annotation exporter generates a stream of data reflecting changes in annotations of specific types that have a direct correspondence in StorySpace structures (e. g., events and plot relations). The data is consumed by the Annotation translator that converts them to corresponding actions in the StorySpace.

## 2.6 SEC Store API

The SEC Store API provides an interface to the Information extractor, the Content classifier & analyser and other parts used outside the SEC. It accesses the DECIPHER database which stores all documents, results of NLP processing, annotation suggestions and users' feedback. The centralised document store enables checking whether a given document was already processed (so that relevant results can be just returned from a stored copy) or an analysis still needs to be applied. Details of the SEC Store API are given in Section 5 discussing integration of the SEC within the whole system.

### 3 Methods

This section discusses algorithms and methods implemented in SEC components responsible for relationship extraction. It links structural descriptions introduced in the previous section to relevant functional specifications. It also prepares the ground for evaluations on domain-specific datasets given in the next section.

Almost all of the techniques discussed in following paragraphs are based on a (semi-)supervised machine learning approach. That is why we briefly summarize characteristics of classification methods employed in the DECIPHER system first. The Similarity engine is a distinct component of the Content classifier & analyser module. It provides a set of text relatedness measures that are introduced in Section 3.2. The last part focuses on recognition of named entities. It characterizes a finite state technology providing an efficient mechanism for identifying entity mentions in texts and examines language resources that were used to populate lists of candidate entities.

#### 3.1 Statistical classification and feature representation

A basis of many software components discussed in the previous section can be expressed as a classification task – a piece of text is assigned to one or more theme categories, it is either relevant or not in a specific context, it expresses or not an influence of a painter, etc. Moreover, the decision on what is a correct category for an instance generally depends on a particular application characterized by data to be processed. For example, an art gallery preparing an exhibition on video games as artistic medium would have a very different view on relevancy from a gallery dealing with an exhibition on an old master.

The above-mentioned reasons motivate an employment of statistical classifiers as primary methods in the SEC. Training of the classifiers mainly involves a use of supervised machine learning (ML) algorithms able to build a classification model from existing data. Relevant SEC components integrate resulting models to classify new data.

To obtain fully comparable results, experimental evaluation presented in the next section uses RapidMiner – a general ML software package implementing a set of algorithms that are evaluated on specific datasets. However, it does not mean that the RapidMiner-based implementation of each particular method is necessary the best option when the software component is to be integrated into a production system. Other solutions can be more efficient, have a suitable licencing policy, be easier to integrate or to incorporate results of other DECIPHER modules. That is why the description of individual packages in the previous section mentions other supporting libraries and existing ML tools employed in the current version of the



SEC. New findings and refined requirements resulting from user trials may lead to changes in next releases of the SEC. A final decision on ML modules integrated into the DECIPHER system will be documented in Deliverable D.4.3.1: DECIPHER semantic annotator.

Even though an integrated classification system can hide this fact, a text to be classified is not directly fed into ML methods. It needs to be processed to form a *feature vector* representing characteristics deemed important for a task in hand. There is a trade-off between complexity of features and efficiency of classification. It can take a significant time to construct feature representation reflecting subtle shades of semantics. Moreover, a number of classification experiments proved that sophisticated representations do not necessarily yield better results. Consequently, standard document classification typically uses a very simple feature representation of texts. It assumes that words are independent of their position in a text. This is known as a bag-of-words representation of documents. It is the primary way employed in our classification experiments reported in the next section (bigrams – pairs of words – and trigrams – triplets – are also used).

As the number of words involved in collections of documents is large, the feature vectors can be lengthy. Moreover, each individual term representing a low level feature of a document usually contributes only a little to an overall classification. To overcome this, researchers proposed various dimensionality reduction strategies. In addition to straightforward feature selection and feature combination approaches, there is a wide range of low dimensional latent representations that aim at better capturing document semantics. For example, a well-known technique of the Latent Semantic Indexing (LSI) [7] maps documents associated with terms onto a latent space by performing a linear projection: singular value decomposition. The Principal Component Analysis (PCA) [21, 1], an alternative to the LSI, projects high dimensional term vectors to a lower dimensional space by finding a solution of an eigenvalue problem. A step forward to statistical models is the Probabilistic Latent Semantic Indexing (PLSI) [15], which defines a proper generative model to sample terms from a mixture distribution. Related statistical models include Latent Dirichlet Allocation (LDA) [3, 2], Generalized Dirichlet Multinomial Distributions (GDMD) [4], or Rate Adapting Poisson (RAP) [11] models. There are also methods projecting a given document to a space given by explicit dimensions defined externally. For instance, Explicit Semantic Analysis (ESA) [10, 12] employs Wikipedia articles for this job.

It should be noted that the above-mentioned techniques are generally computationally intensive as they rely on expensive matrix decomposition algorithms. Scalable alternatives exist, e.g., Random Indexing (RI) [23] minimizes computations by employing an approximation of the expensive orthogonalization of the subspace (by using random matrices). Nevertheless, there is still a performance gap between the needs of components

that call for a fast on-line adaptation (involving re-training) and capabilities of advanced dimensionality reduction procedures. Consequently, the SEC employs the methods when dealing with data that can be processed once and used many times, namely, in the Similarity engine module, but it relies on low level features in standard text classification tasks.

Information extraction components in the SEC take into account only a close neighbourhood of an information unit (a context window, a sentence, a paragraph, etc.) so that the word order and a position in a text play a significant role. Prepositions and other functional words that are usually ignored in text classification tasks provide vital clues in information extraction. Of course, relation mining takes also advantage of named entity recognition.

The reading level estimation employs stylistic features that are derived for established formula published in relevant scientific literature – Flesch’ Reading Easy score [9], the Gunning Fog index [13], the Flesch-Kincaid Grade level [19] and the Coleman-Liau index [6]. Textual characteristics include average lengths of sentences and words, ratios of verbs, nouns, adjectives, and adverbs, and average numbers of words belonging to each individual word frequency quartile (with a log scale). The R-Rating adds a comparison with a list of common vulgar, erotic, violent, or racist words and phrases. The interestingness further extends the set of features characterizing frequencies of words in a specific type of text. For example, the word *vegetarian* is infrequent in the collected biographies of famous painters so that a high interestingness score will be assigned to a sentence *Da Vinci was a vegetarian*.

It should be realized that even a very simple classification method such as the Naive Bayes classifier, which is used as a baseline in comparisons reported in the next section, can have its place in a production system. Even though the methods are consistently outperformed by their advanced counterparts, it makes no sense to employ sophisticated and complex classification techniques when there is only a trickle of training examples provided for a task.

The length of a text to be classified also influences performance of individual classifiers. Results presented in the next section suggest that it would be valuable to train a specific classifier for each “length category” the SEC operates on (e. g., theme classification just on titles of paintings). Moreover, it is often necessary to combine classifiers based on ML principles with components relying on predefined lists of task-dependent words (e. g., to decide whether a painting entitled “Goliath” could refer to a biblical theme). These directions will be addressed in our future work too.



### 3.2 Measures of semantic relatedness

The Similarity engine provides various measures characterizing term and phrase semantic similarity and relatedness. Two sets of resources are involved – manually created and automatically derived ones. The former group consists of large-scale general-purpose resources – the Wordnet<sup>22</sup>, the EAT (Edinburgh Associative Thesaurus)<sup>23</sup> resulting from free word association tests<sup>24</sup>, and Wikipedia alternative names and redirections as well as a domain-specific thesaurus – the Getty AAU Thesaurus<sup>25</sup>. The latter is represented by collections of semantically clustered words derived from textual corpora by means of statistical semantics techniques.

Mechanisms to compute actual scores of the semantic distance differ according to varying character of underlying data. The WordNet::Similarity module<sup>26</sup> [22] is used to compute wordnet similarity measures. The similarity among words found in the EAT is weighted by frequencies of words given as answers to a given stimulus word. Wikipedia alternative names and redirections are simply interpreted as synonymous expressions. The distance of terms appearing in the Getty AAU Thesaurus is computed as an inverted number of links connecting the terms (an artificial root node is added to guarantee connectivity).

Statistical semantics methods are based on the Distributional hypothesis which suggests that words that occur in the same contexts tend to have similar meanings [24]. The underlying idea that “a word is characterized by the company it keeps” was popularized by John R. Firth [8]. A key parameter of the methods therefore lies in a definition of the context.

Results of statistical methods integrated in the DECIPHER Similarity engine are generated in several alternatives that correspond to varying datasets from which they are computed, the context considered as a neighbourhood (a fixed-size context window, a sentence, a paragraph, and a whole document), and the way the data is processed. Large collections of general as well as domain-specific documents were used. The English Gigaword and the Italian itWaC corpora were used for general-language term similarity computations. Domain-specific models were computed from data collected within the DECIPHER project, which include: Freebase<sup>27</sup>, DBpedia<sup>28</sup>, Art.sy<sup>29</sup>, books and other documents relevant to the visual art

---

<sup>22</sup><http://wordnet.princeton.edu/>

<sup>23</sup><http://www.eat.rl.ac.uk/>

<sup>24</sup><http://w3.usf.edu/FreeAssociation/>

<sup>25</sup><http://www.getty.edu/research/tools/vocabularies/aat/>

<sup>26</sup><http://wn-similarity.sourceforge.net>

<sup>27</sup>[http://www.freebase.com/view/visual\\_art](http://www.freebase.com/view/visual_art)

<sup>28</sup><http://dbpedia.org/>

<sup>29</sup><http://art.sy/>

available as a part of the Gutenberg project<sup>30</sup> and Open library<sup>31</sup>, VADS<sup>32</sup>, Art Media Agency<sup>33</sup> and others.

A simple extraction method takes into account only a joint occurrence of terms in a given context window. Thus, it mainly covers syntagmatic relations between words (e.g., *sick* will be closer to *child* than to *ill*) and words typically appearing in conjunctive constructions (e.g., *madonna* and *child*).

An advanced technique considers also words that do not necessarily co-occur in the same context but that appear regularly in the context of both the words. We employ the Second Order Co-occurrence PMI (SOC-PMI) word similarity method [16] that uses the Pointwise Mutual Information to sort lists of important neighbour words. The method considers words which are common in both lists and aggregate their PMI values (from the opposite list) to calculate the relative semantic similarity [17].

The previous two approaches do not take into account distinctions between parts of speech (e.g., they can put together a noun and an adjective) and do not consider similarity given by occurrence in same syntactic relations. Two other implemented techniques benefit from advanced language-specific pre-processing of texts – by means of PoS taggers and syntactic parsers, respectively. The former one simply filters candidate word lists by their PoS so that only nouns can be similar to nouns, verbs to verbs, etc. The latter employs the concept of Lin’s Lexical Semantic Similarity Measure [20] and the Word Sketches [18] and puts together words that often participate in same relations with same words.

Finally, alternatives of the above-mentioned methods were generated using LSA, pLSA, LDA, Random Indexing and ESA techniques. We took advantage of efficient implementations – the Gensim Python tool for experimenting and the Vowpal Wabbit<sup>34</sup> LDA system [2] that has been integrated into the current version of the SEC. This implementation is very fast and can deal with a large number of textual documents.

### 3.3 Finite state machines and extraction patterns

The task of named entity recognition and classification (NERC) can rely on various methods, depending on the nature of entities to be identified. For example, temporal expressions are usually recognized by means of a set of predefined rules that locate relevant parts in a text and transform a found expression to a normalized form. Nevertheless, there is a common part of all NERC approaches – scanning an input text and identifying candidate

<sup>30</sup><http://www.gutenberg.org/>

<sup>31</sup><http://openlibrary.org/>

<sup>32</sup><http://www.vads.ac.uk/>

<sup>33</sup><http://www.artmediaagency.com/en/>

<sup>34</sup><http://hunch.net/~vw/>

words and phrases that could belong to an entity mention. This is true for general as well as domain-specific named entities. The former is typically represented by geographical place names, the latter by names of people, works, etc. Resulting lists of terms to be searched for in input texts can be very large.

To be able to store and match millions words or multi-word expressions corresponding to named entities of potential interest, it was necessary to find an efficient mechanism applicable for the task in the DECIPHER project. After experimenting with various alternatives, we employed a freely available package<sup>35</sup> which implements an incremental method constructing minimal finite state automata (FSA) or transducers from sorted lists of predefined keywords.

A resulting text scanning module is very fast. Table 1 summarizes results of a FSA built from a list of entities of one type. An integrated system that covers all entity types remains efficient – it is able to process more than 20,000 words per second on a standard desktop computer. Moreover, it significantly reduces memory requirements of storing long lists of expressions corresponding to named entities. For example, a resulting representation of all relevant data from the GeoNames database, which originally took more than 1.1 GB, takes only 71 MB in the FSA representation.

length of input in words	size of input	processing time
10,000	64 kB	0.202 s
100,000	640 kB	1.138 s
1,000,000	6.4 MB	10.433 s
10,000,000	64 MB	102.719 s

Table 1: Performance characteristics of the NERC component

Several available software packages recognizing date and time properties of events were tested and compared. HeidelTime<sup>36</sup> – a multilingual temporal tagger was chosen as the best performing tool. The system extracts temporal expressions in natural language texts and normalizes them according to TIMEX3<sup>37</sup> – the ISO temporal annotation standard. HeidelTime is a rule-based system. As the source code and the resources (patterns, normalization information, and rules) are strictly separated, one can easily extend the resource specification. In the TempEval-2 challenge<sup>38</sup>, HeidelTime achieved the highest F-Score (86 %).

<sup>35</sup><http://www.eti.pg.gda.pl/katedry/kiw/pracownicy/Jan.Daciuk/personal/fsa.html>

<sup>36</sup><http://dbs.ifi.uni-heidelberg.de/index.php?id=129>

<sup>37</sup><http://timeml.org/site/timebank/documentation-1.2.html>

<sup>38</sup><http://www.timeml.org/tempeval2/>

The system mainly uses hand-crafted regular expression patterns to extract temporal expressions from natural language texts and knowledge resources as well as linguistic clues for their normalization. Generally, HeidelTime extracts four types of temporal expressions – *Date*, *Time*, *Duration*, and *Set*. An advantage of the system is that one can choose either a better precision or a better recall when annotating temporal expressions in text. The modes should achieve about the same F-Scores. Similarly to other systems, HeidelTime allows users not only to extract temporal expressions but also to normalize them. It allows normalizing of simple temporal expressions, such as *March 11, 1982*, as well as more complicated ones, such as *Independence Day 2010* or *last June*.

The geographical database of Geonames.org was used to populate a list of geographical names to be recognized by the DECIPHER system. It provides freely available lists covering millions of place names around the world. The data contains over 10 million geographical names corresponding to 7.5 million unique geographical features whereof 2.8 million populated places and 5.5 million alternate names. The features include political entities, lakes, parks, populated places etc. The data is accessible via a number of web services as well as via a daily database export. Geonames.org stores names of places in various languages and integrates additional geographical data such as elevation, population and lat/lon coordinates. Users can contribute to improving the quality of data by correcting and adding new names via a wiki interface.

To recognize domain-specific named entities, such as artists, styles, or artworks, various sources were explored and acquired lists of entities were integrated. Wikipedia and Freebase accounted for a vast majority of resulting lists – 1,627,914 entities were identified. Additional resources added only a limited number of new entities – Artcyclopedia<sup>39</sup> – 3,903 entities, SCoT<sup>40</sup> – 1,137, Art.sy<sup>41</sup> – 164, etc.

Following categories from the Freebase database were extracted: visual artist, artwork, material, organization, art form, art movement and event. A set of attributes is defined for each entity: name, normalized name, type, source, URL and visual representation (for visualisation purposes in DECIPHER). Entities extracted from Freebase and SCoT contain also Freebase ID or SCoT ID, respectively.

However large they are, predefined lists of named entities are never complete. A set of heuristics is therefore employed, which enables discovering unknown entities beyond the lists. Surface form clues of words such as capitalization and predefined patterns are employed. Let us consider a sentence *among legendary sculptures, there are such classic pieces as Boy*

<sup>39</sup><http://www.artcyclopedia.com/>

<sup>40</sup><http://scot.curriculum.edu.au/>

<sup>41</sup><http://www.art.sy/>

*With a Basket of Fruit by Caravaggio* and a situation in which Caravaggio is a known artist but the system does not know that *Boy With a Basket of Fruit* is a painting by Caravaggio. The capitalization helps to identify the sequence of words that are likely to form an entity (of an unknown category, yet). Next, a simple pattern *ARTWORK by ARTIST* is matched and the unknown entity is categorized as an artwork.

Patterns form also a basis of relation extraction tasks in DECIPHER. They can be user-defined, but also automatically derived from annotated data. The current version of the Information extraction component is able to identify relations of various kinds of influences, travels and events that are reflected in artworks. Nevertheless, users are able to specify other types of relations (and respective roles) and provide examples of annotations to train the system for a new task.

Even though it is generally possible to use simple regular expressions to define extraction patterns, it is advantageous to perform sentence parsing first, identify dependencies among entities and let extraction rules work on the syntactic level. The DECIPHER Relation extractor thus calls the On-demand NL processor and operates on produced parsed texts with basic co-references resolved (by means of the Stanford Parser and the Stanford Deterministic Coreference Resolution System for English).

The general co-reference resolution does not identify all links (references) necessary for the intended relation extraction in DECIPHER. For example, it is not able to find an influence extraction pattern in the following text: *Another paradigmatical work of this period is the famous "Nevermore". The painting pays tribute to the famous poem by Edgar Allan Poe.* To cope with such situations, the DECIPHER Coreference resolution system was extended by a set of domain-specific rules that are able to link words such *painter, sculpture, poem, style, place*, etc., corresponding to entity types identified by the NERC module, to the closest entity mentioned in the text or inferred from the knowledge base (works by authors, hierarchy of places and so on).

## 4 Evaluation

### 4.1 Relevance classifier

As mentioned in Section 2.3, the Relevance classifier module is used for two tasks in the project. First, it is employed as an input filter reducing processing costs by eliminating documents that are irrelevant w.r.t. to the domain in focus – visual art. The other role the classifier plays consists in identifying documents relevant for a specific task, e.g., finding biographies or sources of art prices. This division is also reflected in experiments reported in this subsection.

The data set for general relevancy experiments consisted from all documents on visual art from Freebase and a subset of Wikipedia pages related to the same subject. Negative examples were taken from other, non-relevant Freebase categories and Wikipedia pages linked from them.

Results of four classification algorithms were compared (the same set of classifiers is used in all other reported experiments):

1. Naive Bayes
2. k-Nearest Neighbours (k-NN)
3. SVM with linear kernels (SVM linear)
4. SVM with radial basis functions as kernels (SVM rbf)

Three values of parameter  $k$  – 1, 3 and 7 – are reported for the k-Nearest Neighbours.

Results of the experiment are shown in Figure 3. It is obvious that the large data allows all methods that generalize in the train phase (all tested classifiers except the k-NN classifier) to achieve good results. The SVM with linear kernels was integrated into the DECIPHER system.

The second set of experiments considered documents that correspond to a specific type of text in the visual art domain. As results on several training sets were similar, only the performance of biography classifiers will be reported here. The task of automatic methods was simply to decide, whether an in-domain text correspond to a biography of an artist or not. One hundred positive samples – real biographies – were collected from <http://www.biographies.com> and other sources. Negative samples were taken from non-biographies in the visual art subset of Freebase in the same quantity. Figure 4 summarizes results. The SVM with linear kernels confirmed its superiority in relevance classification so that it is used for the task in the DECIPHER prototype.

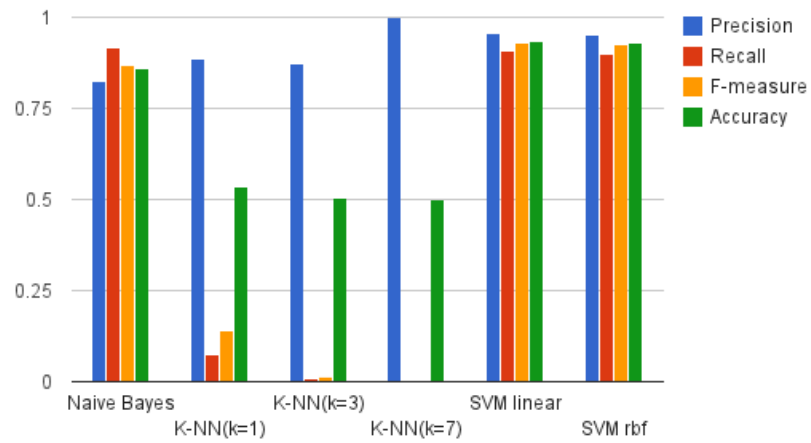


Figure 3: Results of general relevance classification

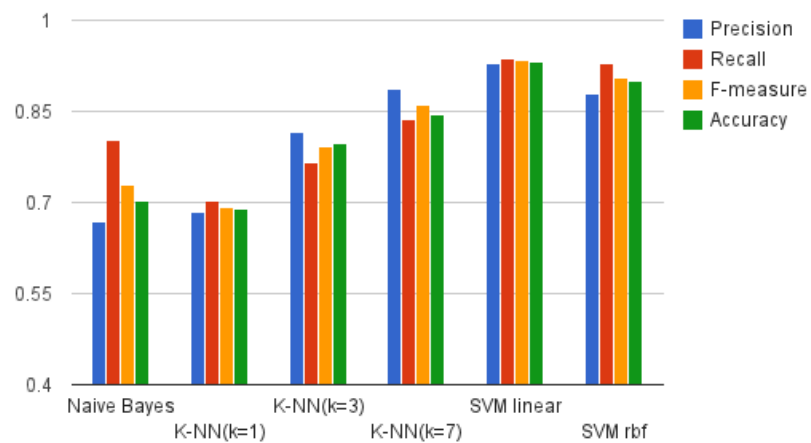


Figure 4: Results of biography classification

## 4.2 Content acceptability estimator

The Content acceptability estimator consists of three interconnected classifiers. An analysis of actual practices of museum professionals led to a change of original categorization of reading levels to specific classes for primary schools, secondary schools, and adults. Training sets for the primary school category were obtained from web pages dealing with “art for kids” such as Tate Kids<sup>42</sup>, NGA for Kids<sup>43</sup>, IvyJoy<sup>44</sup>. The secondary school material was collected from high school gallery programs such as the BBC Art Sites for School Ages 11–16<sup>45</sup> or the National Gallery Secondary School Programme<sup>46</sup>. The training set for adults was taken from the Art Newspaper<sup>47</sup>, the Arts Journal<sup>48</sup>, etc.

Tests were run on 200 documents from each category. Results are shown in Figure 5. As mentioned in Section 2.3, each document is represented by a short list of numbers characterizing an average length of sentences, frequencies of words, etc. That is probably the reason why results of the two SVM modules are almost indistinguishable.

Even a document that uses basic words and a simple style can show to be inappropriate for children as it may contain inappropriate words. The R-Rating module takes advantage of available lists of words with a potentially improper meaning. Some words are inappropriate only in certain contexts so that neighbour words are also considered. The black list contains common vulgar, erotic, violent and racist words, taken mostly from the Dictionary of the Vulgar Tongue<sup>49</sup> and similar web resources.

Articles and books from which testing data was taken were mostly downloaded from the Project Gutenberg<sup>50</sup>. Coverage of the R-Rating component was evaluated in terms of “misses” – inappropriate cases that were not identified – per category of sources. Table 2 summarizes the results. A relative high number of misses in category “Criminal & violence” is mainly given by a strict assessment of testing sentences. Also, sentences from children books are sometimes taken as inappropriate due to violence words.

Finally, the Interestingness classifier is called to identify the most interesting parts (e.g., sentences or paragraphs) in a document. Of course, interestingness is subjective so that the results need to be interpreted in the context of the dataset collected for experiments. One hundred web pages

<sup>42</sup><http://kids.tate.org.uk/>

<sup>43</sup><http://www.nga.gov/kids/>

<sup>44</sup><http://www.ivyjoy.com/fables/>

<sup>45</sup>[http://www.bbc.co.uk/schools/websites/11\\_16/site/art.shtml](http://www.bbc.co.uk/schools/websites/11_16/site/art.shtml)

<sup>46</sup><http://www.nationalgallery.org.uk/learning/teachers-and-schools/secondary-schools/>

<sup>47</sup><http://www.theartnewspaper.com/>

<sup>48</sup><http://www.artsjournal.com/visual.shtml>

<sup>49</sup><http://www.fromoldbooks.org/Grose-VulgarTongue/>

<sup>50</sup><http://www.gutenberg.org/>



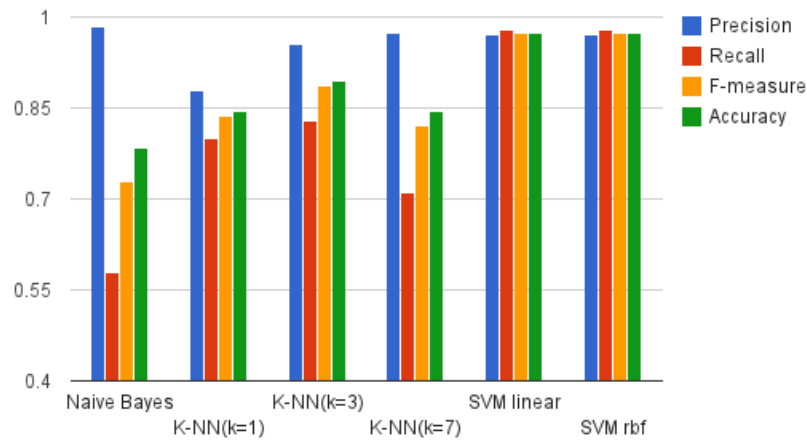


Figure 5: Results of the reading level classifier

Category	Misses/incorrectly recognized
Erotic/porn stories	3.57 %
Horror	18.31 %
Criminal & violence	35.48 %
Children books	6.25 %

Table 2: Results for the R-Rating experiment

were manually annotated and the most interesting parts were tagged. The pages were chosen to mention one of the facts discussed in art fan web sources such as the Obscure Facts About Famous Artists<sup>51</sup>.

Figure 6 characterizes classification results of experiments on the collected dataset. Support vector machines with radial basis functions as kernels provided the best results. However the performance is rather limited. Our future work will focus on collecting more data and implementing advanced features that could bring improvements of the results.

<sup>51</sup>[http://www.huffingtonpost.com/2011/07/18/obscure-facts-about-famous-artists\\_899475.html](http://www.huffingtonpost.com/2011/07/18/obscure-facts-about-famous-artists_899475.html)

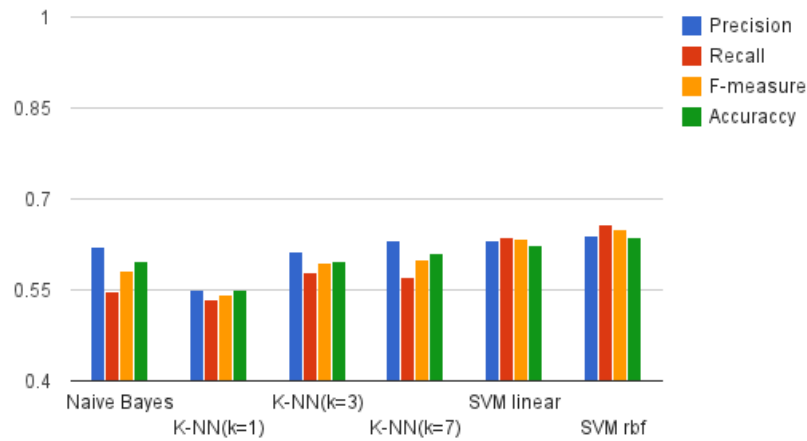


Figure 6: Results of the Interestingness classifier

### 4.3 Theme classifier

A primary motivation of the theme classification in DECIPHER is its integration into the process generating recommendations of potentially related heritage objects. The function can be invoked from a story of an explicitly stated theme but the theme can be also implied from a title of an individual heritage object and supplementary textual data. In any case, there is a clear need for an automatic tool that predicts major theme of heritage objects from texts associated with them.

To estimate performance of automatic methods on data characterizing varying nature of themes and their interrelations, theme classifier tests were run on 6 distinct datasets containing texts associated with heritage objects in the following categories:

- Biblical/Religious
- Mythological
- History
- Landscape
- Portrait
- Genre painting

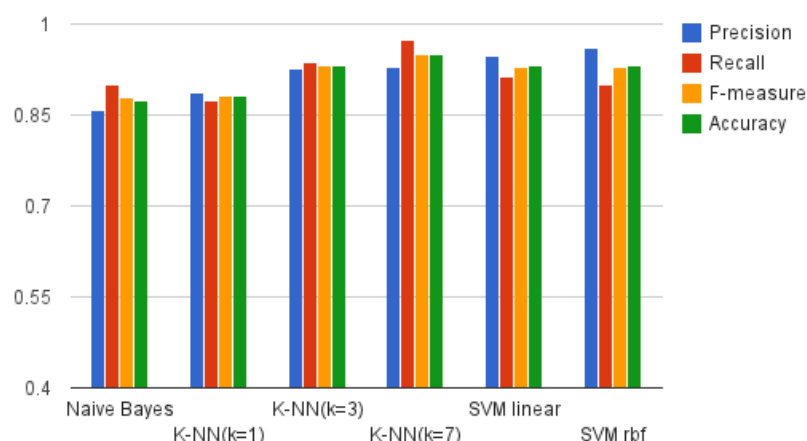


Figure 7: Results of biblical/religious theme classifiers

Obviously, the first two themes are very close (texts on mythological themes that do not belong to the biblical/religious category were used as negative examples for the first dataset). This reflects fine-grained distinctions required for recommendations in the case of preparing exhibition on specific themes. The last three categories correspond to themes of paintings as recognized by standard scholar literature on visual art. It is difficult to generalize characteristics of the broad classes, e. g., variability of titles and other texts on landscape paintings is very high and there are many outliers.

Themes generally overlap and a heritage object may belong to multiple categories. Nevertheless, the experiments were modelled as six binary classification tasks. Scores or certainty factors of the classifiers are available so that mutual exclusion of classes can be realized by taking a classifier with the strongest response.

Figures 7–12 demonstrate performance of the classifiers. Support vector machines with linear basis functions give the best results on history, landscape, portrait, and genre painting categories. Interestingly, k-NN models overcome advanced techniques in the first two categories. This is probably caused by the specificity of the themes – it is advantageous to take just closest examples and classify a new text considering only their decisions.

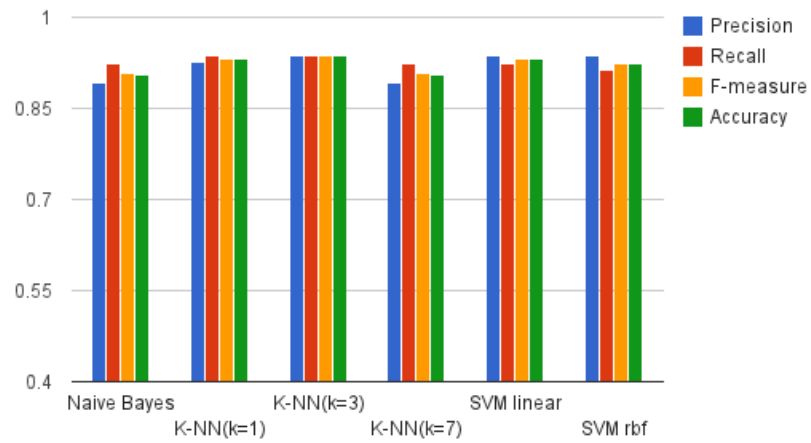


Figure 8: Results of mythological theme classifiers

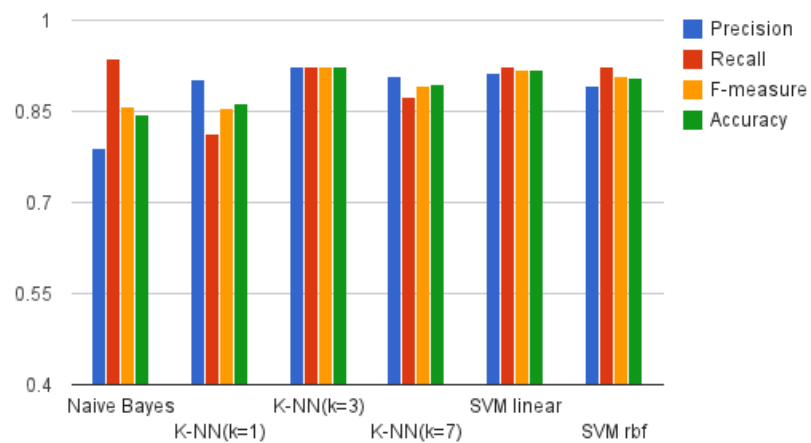


Figure 9: Results of history theme classifiers

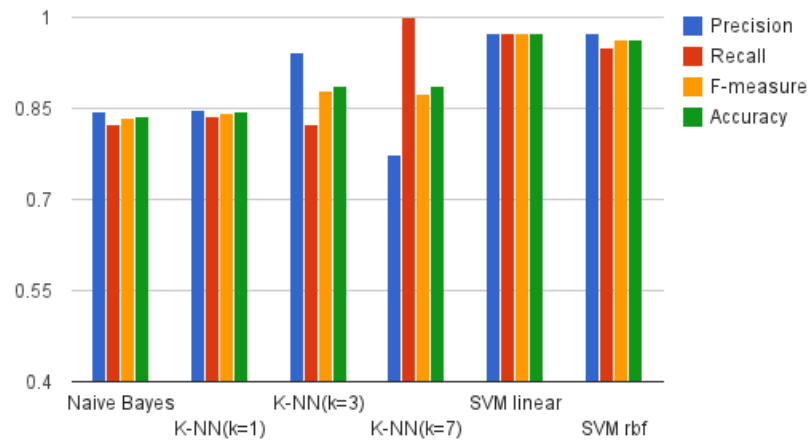


Figure 10: Results of landscape theme classifiers

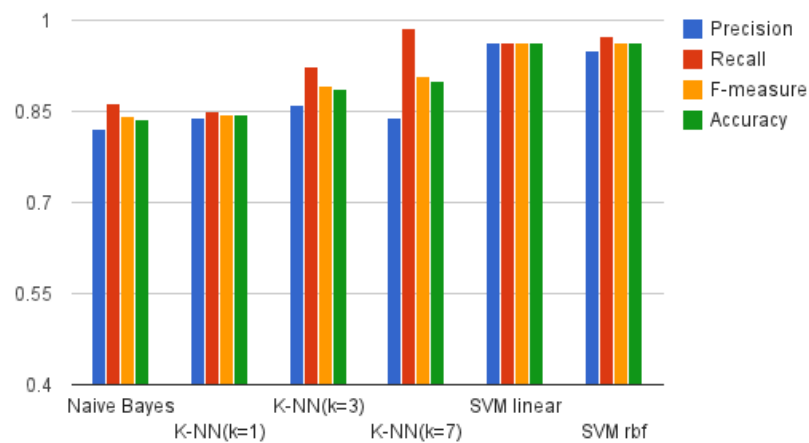


Figure 11: Results of landscape theme classifiers

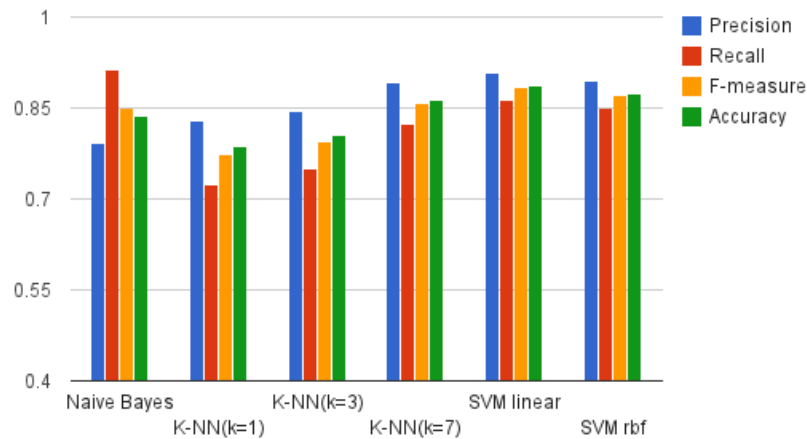


Figure 12: Results of genre painting theme classifiers

#### 4.4 Opinion miner

As mentioned in Section 2.3, the Opinion miner consists from two parts – a sentiment analyser and a controversy identifier. The former is applied to blog reports referring to exhibition visits and users’ feelings about particular artworks. Individual textual fragments as well as whole documents can be classified as *neutral*, *positive* or *negative* (documents can be also marked as *mixed sentiment* if they contain about the same amount of positive and negative parts).

Preliminary tests showed that available sentiment analysers that are trained on data from other domains have a suboptimal performance on the DECIPHER dataset. Thus, we focused on adaptation of an existing classifier developed for a movie recommendation application to the visual art domain and an evaluation of the resulting system on the data collected within the project. The response of the original system was added as a new feature to the data and the system was able to simply accept the previous decision or change it.

More than 200 blog posts were manually annotated. Data was collected from DoodlersAnonymous<sup>52</sup>, ArtThreat<sup>53</sup>, LineSandColors<sup>54</sup>, Invesp<sup>55</sup>, Art

<sup>52</sup><http://www.doodlersanonymous.com/>

<sup>53</sup><http://artthreat.net/>

<sup>54</sup><http://www.linesandcolors.com/>

<sup>55</sup><http://www.invesp.com/blog-rank/Art>

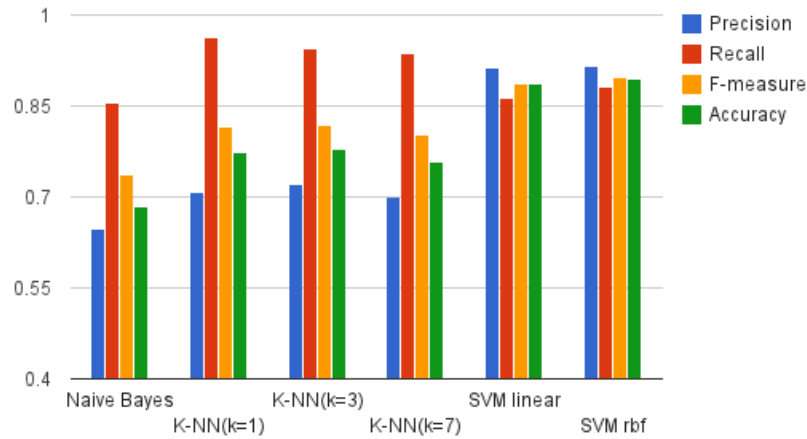


Figure 13: Results of sentiment analysis experiments

FagCity<sup>56</sup>, and other relevant sites. Figure 13 shows results of the experiment. Performance of the two SVM-based classifiers is almost identical and superior to other tested methods.

The second part of the Opinion miner aims at an identification of conflicting views that are explicitly stated in texts. This task can be also seen as a specific case of relation extraction where a relation type corresponds to controversy and attributes are opposite views on an artwork, an artist or an exhibition.

Relevant text documents were collected from the web based on a manually created list of words and phrases expressing polemic statements. Parts corresponding to terms on the list were automatically tagged in the documents and real cases of opposed opinions were manually annotated. There were 244 documents in total in which 77 cases of conflicting opinions were manually identified.

A classifier based on the Conditional Random Fields was trained and evaluated. In a 10-fold cross-validation evaluation experiment on the data, it reached the accuracy of 81.3 % and the F-measure 76.6 %.

#### 4.5 Relation extractor

Two experiments were run to evaluate the Relation extraction component described in Section 2.4. They concentrate on relations of influences that

<sup>56</sup><http://www.artfagcity.com/category/review/>

play a crucial role in DECIPHER (among others, they can be visualised and presented to users as a graph).

A large scale dataset was prepared for the first evaluation experiment. It focuses on a specific subtype of the influence relation – *ARTIST influenced ARTIST* – for which there is a significant amount of existing data which can be used for evaluation purposes. We extracted relations *influenced* (8,862) and *influencedBy* (18,283) from DBpedia and Freebase and merged the data to form a consistent list of triples, e. g.:

*Francisco\_Goya influenced Pablo\_Picasso*

*Anthony\_van\_Dyck influenced Bartholomeus\_van\_der\_Helst*

*Leonardo\_da\_Vinci influenced Peter\_Paul\_Rubens*

The names of artists are not taken as simple strings – they are anchored in the DBpedia so that each person is uniquely identified by a URI (the NERC component was used to normalize all the names first).

Texts in which mentions of two artists linked by the influence relation co-occur were identified in the next step. Wikipedia articles were used as a training source for inferring extraction patterns identifying how the influence relation can be expressed in a text. The rest of matching documents collected within the project were used as a test dataset.

Of course, a co-occurrence of two names of artists in a text does not automatically means that the text expresses an influence relation between the two artists. Nevertheless, this approach enables collecting training documents in a quantity that could not be easily reached by manual tagging. To further increase correctness of the evaluation data, collected documents were further filtered to keep only texts that contain one of trigger terms of influence relations (mostly verbs and verb phrases indicating the relation type). A manual check of a subset of 200 cases proved that the resulting data is accurate – an estimated error rate was 11 % and incorrectly marked documents mainly listed the two artists as influencers of someone else.

Deriving a set of high-coverage extraction patterns is difficult even if one has positive examples from the whole Wikipedia. Table 3 demonstrates high variability of potential contexts on several examples of Wikipedia sentences mentioning an artistic influence between two artists. Incorrect parsing and co-reference resolution further complicate the task.

Extraction patterns were applied to a test set consisting of documents apart from Wikipedia. Results were matched against the prepared “ground truth” data. Table 4 presents accuracy values for the most frequent trigger words. An analysis of misrecognized relations showed that there is a wide range of causes of errors – complex structure of sentences and consequent inferior quality of parsing results, low generality of extraction patterns for less frequent relations, unresolved implicit references, etc. The overall accuracy was 57.8 %. This corresponds to results of other relation extraction systems on similar datasets as summarized in [14]. It also provides



During this period he (coref: Wilhelm Truebner) also made the acquaintance of Carl Schuch, Albert Lang and Hans Thoma, German painters who, like Trübner, greatly admired the unsentimental realism of Wilhelm Leibl.
He (coref: Edward Joseph Ruscha IV) was also impacted by Arthur Dove's 1925 painting Goin' Fishin', Alvin Lustig's cover illustrations for New Directions Press, and much of Marcel Duchamp's work.
He (coref: Micon the Younger of Athens) was closely associated with Polygnotus of Thasos, in conjunction with whom he adorned the Stoa poikile ("Painted Portico"), at Athens, with paintings of the Battle of Marathon and other battles.
He (coref: Gregoire Boonzaier) now absorbed the work of Van Gogh, Cezanne, Utrillo and Braque, and made trips as far afield as Russia, where his socialist leanings are reinforced.
Assteyn's compositions are particularly affected by the Johannes Bosschaert, and his painting technique is reminiscent of Balthasar van der Ast.
He (coref: Teodoro Duclère) is one of the painters considered to belong to the School of Posillipo that arose in Naples associated with the Dutch painter, Anton van Pitloo, who would become Duclères father-in-law.
He (coref: Francesco Maggiotto) continued painting in the style of his father, hence his nickname, but he could not fail to be influenced by the last generation of great Venetian painters, from Tiepolo to Pietro Longhi.
Chesley Bonestell, considered by many to be one of the most accomplished practitioners of the space art genre, critiqued Davis early paintings and encouraged him to pursue an artistic career.
Eakins married Susan Hannah Macdowell, one of his students at the Academy, in 1884. ... Unlike many, she was impressed by the controversial painting and she decided to study with him at the Academy, which she attended for 6 years, adopting a sober, realistic style similar to her teacher's.
Since her childhood, she had been interested in the arts, specially painting, being a disciple of Fernando Alvarez de Sotomayor and Pablo Burchard
Wellington was regarded as a pictorial photographer of note, while his work was clearly inspired by the paintings of John Constable and Thomas Gainsborough.
Born in 1954 he (coref: Go Arisue) became interested in rope bondage at an early age inspired by paintings of the legendary bondage artist Seiu Ito and the works of Akira Minomura.
He (coref: Eduardo Afonso Viana) latter followed the post-impressionist style inspired by Cézanne in some of his best paintings.

Table 3: Examples of influence mentions from Wikipedia

Trigger	Frequency	Accuracy
influenced	1,149	58.6 %
was teacher of	622	58.2 %
inspired	496	55.4 %
encouraged	211	57.8 %
appreciated	153	56.9 %
admired	138	57.2 %

Table 4: Results of relation extraction – influence of one artist on another one

a good basis for integration into the component generating annotation suggestions.

The second dataset for evaluating automatic extraction of influence relations focused on fine-grained subcategorization of relation attributes. The data was manually annotated (including validations and potential corrections of co-reference resolution). A part of the dataset was processed by two annotators to evaluate an inter-annotator agreement. Cohen’s kappa coefficient [5] reached 79.8 %. A majority of disagreement was observed in annotations of events (influencing artworks or artists) and influences of groups of artists (such as artistic schools or styles) and their applicability to individual representatives of the groups.

Table 5 lists patterns that were considered in the experiment together with numbers of their occurrences in the dataset (numbers of sentences that were annotated). The 10-fold cross-validation scheme was applied. An average accuracy of the Information extractor reached 56.3 % on the dataset.

Pattern	Frequency
EVENT influenced ARTWORK/ARTIST	143
ARTWORK/ARTIST influenced ARTWORK/ARTIST	126
ARTMOVEMENT influenced ARTWORK/ARTIST	115
PLACE influenced ARTWORK/ARTIST	107

Table 5: Characteristics of the second dataset on influence relations

## 5 Integration with other project components

The SEC Store API defines HTTP-based interfaces for internal components of the SEC such as the Annotation server as well as for other DECIPHER modules – the Aggregator and the StorySpace. The Aggregator can request a batch processing of crawled documents. The SEC Store API then stores the documents, initiates their processing and takes care of results. The StorySpace invokes searches for related documents based on a given seed. It can also query results of automatic classification processes.

The Annotation (4A) server takes also advantage of HTTP interfaces to communicate with three components – the Annotation editor, the Annotation translator, the SEC Store API. Another graphical user interface is used for configuration. Annotation editors, located in the client side of the StorySpace, employ the 4A protocol – a special annotation interchange protocol extended for DECIPHER purposes – to access the Annotation server. The communication is initiated by an editor and, after it, it becomes bidirectional asynchronous so that the server can inform all clients about changes, e. g., about new annotations.

The Annotation translator receives data from the Annotation server and transforms it into StorySpace internal data structures. The translation interface is universal – it can export annotations and send them to any external module. In the current setting, the Annotation translator is passive, i. e., it fully depends on the Annotation exporter which uses the push method to inform the client about new relevant annotations relevant to the StorySpace.

The following subsections detail component interaction patterns in two specific use cases.

### 5.1 Annotation use case

The annotation process starts by invoking the Annotation editor in the Edit mode of a standard StorySpace text editor. After connecting, the Annotation editor communicates with the Annotation server through the 4A HTTP interface (to deal with browser security restrictions, there is also a simple proxy server in the StorySpace, which resends data to the 4A Server and processes results).

Users can ask for annotation suggestions by clicking on the Suggest button. The Annotation editor sends a request to the Annotation server which checks whether annotations for a given document have been already generated and stored by means of the SEC Store API. If it is not the case, relevant SEC components are called to generate suggestions.

As soon as suggestions are ready, they are sent back to the Annotation server and transferred to the Annotation editor which displays them. Users simply reject or confirm proposed annotations, edit suggested annotations

or create completely new ones. Resulting annotations are sent to the Annotation server which stores them to enable re-training of machine learning models that are behind the suggestion generation process.

New annotations can also go to the Annotation translator (through its HTTP interface). A selection of types and sources of annotations to be sent are specified when the translator is registered in the server. If the Annotation translator is temporarily unavailable, the Annotation server waits and sends new annotations only after re-establishing the link to the translator. The annotations are transformed to StorySpace internal data structures. A relevant UML diagram is shown in Figure 14.

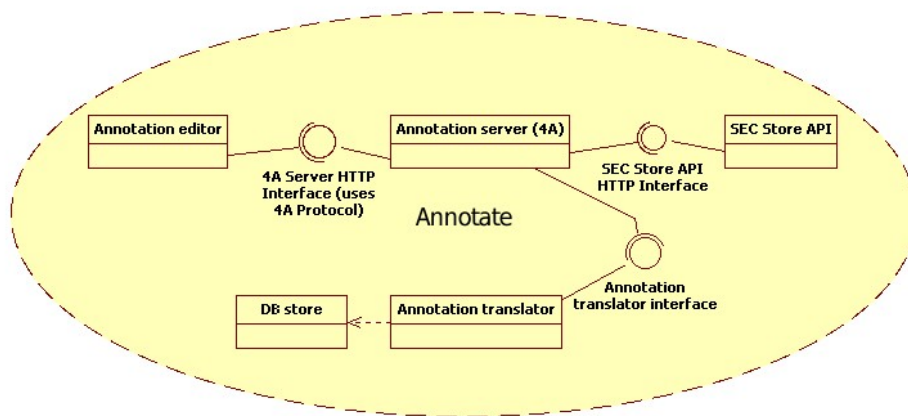


Figure 14: UML diagram of the use case “Annotate”

## 5.2 Search semantically enriched documents

The Search and exploration UI allows users to search data which could be added to the StorySpace. In addition to a simple fulltext search, advanced queries enable specifying constraints on annotations of documents and metadata that was automatically generated by the Content classifier & analyser or by the Information extractor components. The Query and search takes advantage of the Aggregator client which calls the Aggregator API through its interface. The Asset management is used to get actual data. It accesses the SEC Store API through its interface and searches for data corresponding to a query. If it is found, the data is returned to the UI.

If no data corresponding to a given query is available, the Asset management can invoke the Focused crawler to get relevant documents from the web. As crawling and processing web data can take a long time (from minutes to hours), the user is simply informed that the crawling process was initiated and that data will be ready later on. If potentially relevant

documents are found by the Focused crawler, the Asset management is informed. It can initiate semantic enrichment processes by means of calling the SE Scheduler. The Focused crawler can be also directly invoked by a user if currently available data is deemed insufficient. Figure 15 shows a UML diagram of described processes.

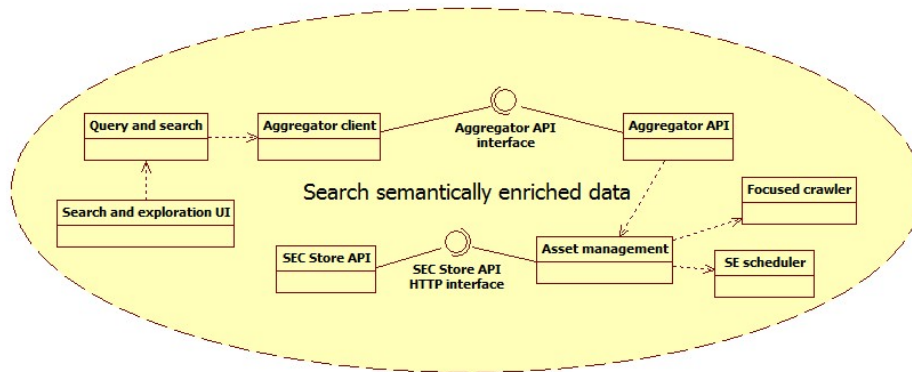


Figure 15: UML diagram of the use case "Search semantically enriched data"

## 6 Conclusions and future directions

The SEC modules discussed in this deliverable provide necessary functionality for automatic identification of various relations. The experimental evaluation proves that there is a strong potential in integrating these results into user interaction components in DECIPHER.

Of course, performance figures measured on collected evaluation data are not sufficient. A real added value of the advanced text mining components will be assessed in user trials in next periods. In particular, merits of the designed and implemented system integrating results of automatic extraction procedures as annotation suggestions will be evaluated in experiments focusing on real user tasks. A preliminary feedback indicates that even if accuracy is limited, as in the case of influence extractions from texts, annotation suggestions help users to easily achieve their goals.

Annotating data for training purposes is tedious so that users try to avoid it as much as possible. Another direction of our future work will thus focus on determining a minimal size of training data necessary for acceptable performance of relation extractors. Semi-supervised methods and active learning will be explored to take a maximal advantage of in-domain data collected within the project.

Last but not least, integration with other two major building blocks of the DECIPHER project – the StorySpace and the Aggregator – will be further improved. This will primarily concern translating annotations into the internal representation of the StorySpace as well as advanced functionality of the SE scheduler.

## References

- [1] ABDI, H., AND WILLIAMS, L. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* 2, 4 (2010), 433–459.
- [2] AGARWAL, A., CHAPPELLE, O., DUDIK, M., AND LANGFORD, J. A reliable effective terascale linear learning system. *CoRR abs/1110.4198* (2011).
- [3] BLEI, D. M., NG, A. Y., AND JORDAN, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3 (Mar. 2003), 993–1022.
- [4] BOUGUILA, N. Clustering of Count Data Using Generalized Dirichlet Multinomial Distributions. *Knowledge and Data Engineering, IEEE Transactions on* 20, 4 (2008), 462–474.
- [5] COHEN, J. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1 (1960), 37.
- [6] COLEMAN, M., AND LIAU, T. A computer readability formula designed for machine scoring. *Journal of Applied Psychology* 60, 2 (1975), 283.
- [7] DEERWESTER, S., DUMAIS, S. T., FURNAS, G. W., LANDAUER, T. K., AND HARSHMAN, R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41, 6 (1990), 391–407.
- [8] FIRTH, J. R. A synopsis of linguistic theory 1930-55. In *Studies in Linguistic Analysis (special volume of the Philological Society)*, vol. 1952-59. The Philological Society, Oxford, 1957, pp. 1–32.
- [9] FLESCH, R. *The Classic Guide to Better Writing: Step-by-Step Techniques and Exercises to Write Simply, Clearly and Correctly*. Signet, 1963.
- [10] GABRILOVICH, E., AND MARKOVITCH, S. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of The 20th International Joint Conference on Artificial Intelligence (IJCAI)* (2007).
- [11] GEHLER, P. V., HOLUB, A. D., AND WELLING, M. The rate adapting poisson model for information retrieval and object recognition. In *Proceedings of 23rd International Conference on Machine Learning (ICML 2006)* (2006), ACM Press, p. 2006.
- [12] GOTTRON, T., ANDERKA, M., AND STEIN, B. Insights into Explicit Semantic Analysis. In *20th ACM International Conference on Information and Knowledge Management (CIKM 11)* (Oct. 2011), B. Berendt, A. de Vries, W. Fan, C. Macdonald, I. Ounis, and I. Ruthven, Eds., ACM, pp. 1961–1964.

- [13] GUNNING, R. *The technique of clear writing*. McGraw-Hill New York, NY, 1968.
- [14] HOBBS, J. R., AND RILOFF, E. Information Extraction. In *Handbook of Natural Language Processing, Second Edition*, N. Indurkha and F. J. Damerau, Eds. CRC Press, Taylor and Francis Group, Boca Raton, FL, 2010, ch. 21.
- [15] HOFMANN, T. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (New York, NY, USA, 1999), SIGIR '99, ACM, pp. 50–57.
- [16] ISLAM, A., AND INKPEN, D. Second order co-occurrence PMI for determining the semantic similarity of words. In *Proceedings of the International Conference on Language Resources and Evaluation* (Genoa, Italy, May 2006), pp. 1033–1038.
- [17] ISLAM, A., AND INKPEN, D. Semantic similarity of short texts. In *Recent Advances in Natural Language Processing V*, N. Nicolov, G. Angelova, and R. Mitkov, Eds., vol. 309 of *Current Issues in Linguistic Theory*. John Benjamins, Amsterdam & Philadelphia, 2009, pp. 227–236.
- [18] KILGARRIFF, A., RYCHLY, P., SMRZ, P., AND TUGWELL, D. *The Sketch Engine*. Oxford University Press, 2008, pp. 297–306.
- [19] KINCAID, J. P., (JR), R. P. F., ROGERS, R. L., AND CHISSOM, B. S. *Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel*. Chief of Naval Technical Training, Naval Air Station Memphis, 1975.
- [20] LIN, D. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning* (1998), Morgan Kaufmann, pp. 296–304.
- [21] PEARSON, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine* 2, 6 (1901), 559–572.
- [22] PEDERSEN, T., PATWARDHAN, S., AND MICHELIZZI, J. Wordnet : Similarity - measuring the relatedness of concepts. In *AAAI* (2004), pp. 1024–1025.
- [23] SAHLGREN, M. An Introduction to Random Indexing. *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005* (2005).



- [24] WEAVER, W. Translation. In *Machine Translation of Languages*, W. N. Locke and A. D. Boothe, Eds. MIT Press, Cambridge, MA, 1949/1955, pp. 15–23. Reprinted from a memorandum written by Weaver in 1949.