

APE_x



Deliverable

Project Acronym: APE_x
Grant Agreement number: 297355
Project Title: Archives Portal Europe network of excellence

D6.6 Second analysis report: Applying Web 2.0 solutions in archival applications

Revision: [Final]

Authors:

Kuldar Aas (NAE) Go Sugimoto (NANETH)
Silke Jagodzinski (BA) Stefan Papp (BA)
Zoltan Lux (NAH) Aadi Kaljuvee (NAE)
Mattias Djupdahl (RA)

Project co-funded by the European Commission within the ICT Policy Support Programme



Dissemination Level

P	Public
---	--------

REVISION HISTORY AND STATEMENT OF ORIGINALITY

Revision History

Revision	Date	Author	Organisation	Description
0.1	22.07.2014	Kuldar Aas, all	NAE, all	First full version based on individual contributions from everybody
0.2	26.08.2014	Tõnis Tärna	NAE	WPL review
0.3	29.08.2014	Kuldar Aas	NAE	Updates and changes based on the WPL review
0.4	02.10.2014	Kerstin Arnold	BA	TC review
0.5	21.10.2014	Kuldar Aas	NAE	Updates based on TC comments and comparison with D6.8 (use cases synchronisation)
0.6	28.10.2014	PB	All	PB review
1.0	29.10.2014	Tõnis Tärna	NAE	Final update based on PB review

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. It reflects only the authors' views and the European Community is not liable for any use that might be made of the information contained therein.

TABLE OF CONTENTS

1. INTRODUCTION	6
2. COMPONENTS OF TAGGING AND LINKED OPEN DATA	8
3. MANAGING SEMANTIC ASSETS.....	10
3.1. SCENARIOS FOR ACHIEVING SEMANTIC INTEROPERABILITY IN THE ARCHIVES PORTAL EUROPE.....	10
3.2. INTEROPERABILITY UPDATES OF CURRENT APE PROFILES	13
3.3. CENTRALLY RECOMMENDED SEMANTIC ASSETS IN THE ARCHIVES PORTAL EUROPE	19
3.4. TOOLS AND METHODS TO SUPPORT SEMANTIC ASSETS IN THE ARCHIVES PORTAL EUROPE	22
3.5. USE CASES FOR MANAGING SEMANTIC ASSETS	23
4. NAMED ENTITY RECOGNITION.....	26
4.1. EVALUATING NAMED ENTITY RECOGNITION TOOLS.....	26
4.2. USE CASES FOR NAMED ENTITY RECOGNITION TOOLS.....	28
5. TAGGING	29
5.1. RELATION TO DELIVERABLE D6.1	29
5.2. COMBINING USER TAGGING AND NAMED ENTITY RECOGNITION	29
5.3. REUSING USER-GENERATED TAGS FOR LINKED DATA	30
5.4. USE CASES FOR USER-TAGGING.....	30
6. PERSISTENT IDENTIFIERS	33
6.1. RATIONALE	33
6.2. CRUCIAL CHARACTERISTICS OF IDENTIFIERS	33
6.3. SCENARIOS FOR CREATING IDENTIFIERS	35
6.4. OBJECTS TO BE IDENTIFIED	36
6.4.1. <i>Currently identified entities in EAD, EAG and EAC-CPF</i>	36
6.4.2. <i>Missing entities</i>	38
6.5. USE CASES - USING PID FOR ARCHIVES PORTAL EUROPE FUNCTIONALITY	38
6.6. USE CASES - MANAGING PID IN THE SYSTEM	43
6.7. SUMMARY OF PID ISSUES	45
7. LINKED OPEN DATA	49
7.1. OVERVIEW OF ISSUES	49
7.2. OBJECTIVES FOR ARCHIVAL LINKED OPEN DATA	51
7.3. PREPARING YOUR DATA FOR PUBLISHING AND LINKING IT	51
7.4. ARCHIVAL RDF MODEL	52
7.5. PUBLISHING LINKED OPEN DATA AT THE ARCHIVES PORTAL EUROPE.....	53
7.6. LOD USE CASES.....	54
8. SUMMARY AND OUTLOOK	56
ANNEX I: PAPER EVALUATION OF NAMED ENTITY RECOGNITION TOOLS.....	58
ANNEX II. CLOSER EVALUATION OF NAMED ENTITY RECOGNITION TOOLS.....	71
EVALUATION RESULTS FOR GATE	71
EVALUATION RESULTS FOR STANFORD NLP	75
EVALUATION RESULTS FOR TOOL UIMA.....	85

EVALUATION RESULTS FOR TOOL NERD 91

LIST OF FIGURES

Figure 1: YEAH! Linked Open Data implementation workflow	50
Figure 2: UIMA general architecture (image source: uima.apache.org)	87

LIST OF TABLES

Table 1: Elements currently controlled in ape standards	13
Table 2: Elements and attributes supporting semantic references	15
Table 3: Elements recommended for central semantic control	19

1. Introduction

One of the main aims of the APE_x project is to develop the Archives Portal Europe¹ into an environment, which allows users to find, manage, reuse and share information relevant to him or her in the best possible way.

In order to determine the actual need for such user-oriented functionalities a survey was carried out during the first year of the project (June – September 2012). Based on the information received through the survey a set of core functionalities were identified, analysed in detail and described in the form of business use cases. These actions were also described in deliverables D6.1 (*First analysis report: Applying Web 2.0 solutions in archival applications*) and D6.4 (*1st report on use cases and usability requirements*).

However, out of the five core areas identified by the survey described in D6.1 two were not analysed in sufficient detail due to the need for additional research – tagging and Linked Open Data (LOD). Therefore the main scope of this deliverable is to extend on these two areas and especially outline the state of the art, possibilities and needs in these areas.

It is also worth to note that due to the changes in the general project setup, which occurred during 2013, the uptake of the functionalities described in D6.1 and D6.4 has not yet finished. As such it is also not expected that it would be possible to implement the necessary functionality to support tagging and LOD in the portal during the lifetime of the project, ie until end of February 2015. As a result the scope of this deliverable is more about raising awareness around tagging and linking data in European archives and less about implementation recommendations (as it was planned initially). Nonetheless, the use cases included in this deliverable should form a solid enough basis for the future Archives Portal Europe Foundation to continue the work on these functionalities.

This deliverable consists of 8 chapters and two annexes:

- The current chapter 1, Introduction, provides a basic introduction to the document;
- Chapter 2, Components of tagging and Linked Open Data, describes how these two technologies could be split into more detailed components for a more reasonable analyses of these;
- Chapters 3 - 7 describe in more detail some of the relevant background technologies (managing semantic assets, named entity recognition, persistent identification) and describe the application of tagging and LOD based on these. Each of the chapters concludes also with relevant use case definitions;
- Chapter 8, Summary and outlook, provides a summary of the discussions as well as concise recommendations for further work in the Archives Portal Europe on tagging and LOD;
- Annex I provides a list of named entity recognition tools, which the APE_x project has identified;

¹ <http://www.archivesportaleurope.net/> (viewed: 2 October 2014).

- Annex II provides more detailed evaluation reports of the named entity recognition tools, which seem to be most relevant for the portal.

2. Components of tagging and Linked Open Data

As already shown in D6.1 both the possibility to carry out tagging by the portal's users and LOD are not singular functions but include aspects from many different research and technology fields, most importantly both have a strong relation to and need for semantic technologies.

Key problem areas of tagging have already been discussed in some detail in the deliverable D6.1². As a summary we can state that the main technological questions in user tagging are whether we should tag archival resources in an open way (ie users use the tags they want to) or should it be based on some formal vocabularies or ontologies. As well, as user tagging is very much dependent on the motivation of users we need to think whether and how the task could be organised to be as simple and appealing as possible.

In LOD the main challenges arise next to the "Linking" part of it. Namely, to be able to link archival records to each other we need to be able to define the common parts of archival descriptions which might come in many different languages; also different name forms or terms for the same things might have been used.

As such we can see that both user tagging and LOD can be very much complementary – when we apply user tagging based on some formal or semi-formal semantic assets (vocabularies and ontologies) we can use these same tags to create links between the archival records and their descriptions. Furthermore, we can witness the growing maturity and availability of tools for natural language processing which can be used to simplify user tagging (eg by providing automatic recommendations which can be confirmed by users with a simple "yes" or "no").

As a summary, we have split up the two focus areas to five specific topics, which are also discussed in appropriate chapters below:

- **Managing semantic assets in the Archives Portal Europe:** when we intend to do some level of controlled tagging and also link archival records to each other the most appropriate vocabularies and other semantic assets for the Archives Portal Europe need to be selected. As well, some level of managing functionality needs to be available inside the Dashboard. These aspects are discussed in chapter 3 of this deliverable;
- **Natural language processing and named entity recognition:** while the level of support can vary very much for different languages there are already now some rather reasonable tools and methods available. Especially the field of named entity recognition and appropriate tools (eg tools which are used to automatically find certain terms from more extensive descriptions) is of interest for the Archives Portal Europe. The use of these tools and an evaluation of the most promising ones is delivered in chapter 4 of this deliverable with further details in annexes I and II;
- **Linking recognised entities to semantic assets through user-oriented tagging:** as a continuation of chapter 4 we discuss in chapter 5 how to use the portal's users (ie carry out crowdsourcing) in order to verify the entities which have been automatically identified by the named entity recognition tools. As well, possibilities for connecting the entities to the

² Available at http://www.apex-project.eu/images/docs/D61_Web20_In_Archival_Applications.pdf (viewed: 2 October 2014).

core semantic assets to achieve linking of archival records described in Archives Portal Europe are discussed;

- ***Creating and maintaining a Persistent Identifier infrastructure:*** one of the core needs for LOD is also the availability of a Persistent Identifier infrastructure. Chapter 6 is therefore discussing the practical needs, problems and possibilities of persistent identification;
- ***RDF modelling and the creation of archival LOD inside the Archives Portal Europe:*** finally, chapter 7 is bringing together the background discussions from the previous chapter to describe a possible way for maintaining and publishing LOD from the Archives Portal Europe.

3. Managing semantic assets

The key objective of the Archives Portal Europe is to provide “... access to information on archival material from different European countries as well as information on archival institutions throughout the continent.”³

As of autumn 2014 this objective is well on the way to be achieved as the portal provides central access to more than 50 million descriptive units with 683 archival institutions from 30 European countries being connected.

Still, the archival descriptions available in the portal serve more as “isolated islands”. This means that a user is able to search for descriptive units based on a simple keyword search and using some limited faceted search possibilities but the full potential of further linking the data within and outside the Archives Portal Europe has not yet been exploited. Especially the amount of different languages used and also the changes in place and person names contribute to the fact that simply using a search term does not provide users with all the relevant hits.

As an example – when searching for information on the city of “Berlin” then currently users need to execute multiple searches for multiple name forms (Berlin, Berliin, Berliini, Berlijn, Berline, Berlino etc.) to find all the information in different languages. In addition search results might also include hits on descriptive units, which describe the composer “Irving Berlin” and are irrelevant for the user.

The situation might be improved by adding a semantic interoperability⁴ layer to the current data and the mark-up description standards (apeEAD, EAG2012, apeEAC-CPF). Simply put – instead of giving access to simple text descriptions there could be the possibility to enrich these descriptions with references to semantic assets⁵ – controlled ontologies and vocabularies. If these references are used coherently throughout all the data it becomes possible to link name forms or terms in different languages as well as distinguish between the same words in different contexts (ie Berlin as a place name or a person name). In turn, these vocabularies and ontologies and links between these would allow the Archives Portal Europe to offer more reasonable queries, supports LOD and could provide a backbone for controlled user tagging.

3.1. Scenarios for achieving semantic interoperability in the Archives Portal Europe

While the idea of semantic interoperability introduced above might sound simple then the implementation of it is far from it and requires huge effort from all parties. When talking about semantic assets in the Archives Portal Europe we can define these as “all ontologies, keyword lists, classifiers, etc. which are used to identify, control and define the content of archival descriptions (ie

³ <http://www.archivesportaleurope.net/> (viewed 2 October 2014).

⁴ Semantic Interoperability enables systems to combine [received] information with other information resources and to process it in a meaningful manner (https://joinup.ec.europa.eu/asset/page/practice_aids/what-semantic-interoperability, viewed 2 October 2014).

⁵ “Semantic assets, [on the other hand,] deliver a central terminology to ensure that data elements are interpreted in the same way by communicating parties. (https://joinup.ec.europa.eu/help_topics#19792, viewed 2 October 2014).

description elements within EAD/EAG/EAC)". Therefore these assets allow the according description elements to be reused, managed and searched in a manner, which is semantically interoperable.

For the purposes of the APE_x project we have identified two generic ways for achieving this goal.

1. Central standardisation of semantic assets

Already now Work Package (hereafter WP) 4 is making good effort to enforce semantic interoperability through the work on APE profiles of EAC-CPF, EAD and EAG. Following the definitions above we can see that some semantic assets have already been implicitly enforced, as certain elements in these profiles are covered by rules regarding syntactic (eg support for a specific date format) or semantic interoperability (eg elements describing creator types are only allowed to be filled with defined values). Therefore we can see that one of our possibilities is to deepen the standardisation and involve even more elements, which are controlled in the sense of syntax and/or semantics.

Still the major problem associated with this approach is that it is not practical to go "into the extreme" and standardise too many description elements. The simple reason is that this would significantly increase the work required before uploading information into the Archives Portal Europe. Especially smaller archival institutions are not expected to have the capability of extensive data mapping for elements including place and person names or even topic keywords to an extensive central ontology or keyword list.

Another issue is that while this approach would allow standardising a certain element inside the archival description there might be situations where some of the potentially linked information is residing within more extensive (narrative) elements – inside a half page description of an archival record multiple place or person names might be mentioned. In such cases the element-level standardisation is not sufficient and the possibility of inline tagging (marking just a few words and creating a link to the appropriate semantic asset) would still be needed.

However, central semantic standardisation of key elements in ape standards would allow most control over data and therefore we recommend, in more practical terms, to:

- "Start small" with selecting only a few elements for semantic standardisation and continue enforcing control to additional elements later, thus gradually growing the general level of semantic control and giving data providers to also take up the according semantic assets in their own catalogues;
- It could be one of the future tasks in WP4 or the Archives Portal Europe Foundation (APEF) to double-check the list of apeEAD, apeEAC-CPF and EAG elements with standardisation in mind and come up with a proposal of what would be reasonable to implement in short, mid and long term;
- An alternative could also be to have some elements supporting a central control list / semantic asset but not having it as mandatory. In other words - some elements in ape standards might include "recommended" keyword lists or ontology values (ie like type of record = sound, text, image, etc.) but it would not be mandatory to follow these in case the archival institution does not have the knowledge or resources to undertake the necessary data mapping. However, in this case the ape standards would also need to be updated to highlight specifically whether the semantic asset has or has not been used in the archival

description (ie is the value “Berlin” used as an ontology term or is it simply the value derived from the original catalogue).

2. Semantic mapping environment

There is also another, conceptually different, possibility towards semantic interoperability would be to develop principles and tools for a “semantic mapping environment” next to the current portal.

According to this scenario the following steps should be undertaken:

- The Archives Portal Europe standards would be amended to support links to semantic assets. As an example - the apeEAD would allow to add the information that the element <subject> (= keyword) has been filled with the value “private correspondence” and the value originates from the “National Archives of Tuvalu ontology of subject keywords”.

***Note:** As of now some EAC-CPF elements already include the @vocabularySource attribute, and the newly introduced apeEAC-CPF accordingly supports this, but this approach should/could also be extended to form a generic principle and solution among all standards and elements; currently, the general EAD2002 schema provides the attribute @source for the elements, which may contain controlled vocabularies, but it is so far not made available in apeEAD.*

- The Archives Portal Europe would allow content providers to upload these semantic assets in a predefined technical format like SKOS⁶. In the sense of the previous example - along with the archival description the subject keyword ontology is uploaded to the Dashboard as well;
- The Archives Portal Europe Dashboard would be updated to include (semi)automated tools, which allow institution and country managers to map their semantic assets among themselves or to some central assets provided by the portal. As an example - the institution manager of “National Archives of Tuvalu” would be able to create an automatic mapping of its own subject keyword list to the “Archives Portal Europe subject keyword ontology” by using Google Translate and can then simply browse through the generated mappings and approve or reject these.

In addition the environment could later also be amended to allow for (semiautomatic) tagging of previously uncontrolled elements, similar to what is done by the DBpedia Spotlight⁷ tool and/or crowdsourcing of element tagging confirmation or even ontology mapping.

The “semantic mapping environment” solution would in general be more flexible as it looks at semantic assets as being something on top of the description standards and not built-in. As such it would allow changing flexibly the core ontologies recommended by the Archives Portal Europe without needing to update the description standards (apeEAD, apeEAG, apeEAC) extensively.

However, there are two major issues associated to this:

⁶ Simple Knowledge Organization System, <http://www.w3.org/2004/02/skos/> (viewed 2 October 2014).

⁷ <https://github.com/dbpedia-spotlight> (viewed 2 October 2014).

- According to a survey carried out by WP4 the use of semantic assets in archival institutions is low. Only a few have implemented vocabularies or ontologies for the content descriptions (ie keywords, person and place names, etc.), most semantic assets are used only to control the technical characteristics (as an example the size and type of an archival record);
- The development of the semantic mapping environment is not a trivial task and in a desktop survey we were not able to find any tools, which could be implemented within the Archives Portal Europe Dashboard without major updates.

Therefore the main practical recommendation could be to start looking into a mixed approach of both presented scenarios by starting to look into generic possibilities for semantic mark-up in ape standards. Within this work it might be reasonable to look at CIDOC-CRM⁸ and EDM⁹ as these could already now be regarded as “semantic description standards” which is not the case for apeEAD, apeEAC and EAG2012.

In parallel the functional setup of the semantic mapping environment could be discussed in more detail to be implemented in possible follow-up projects either by the Archives Portal Europe or other interested memory institutions and portals (like Europeana).

Finally we would also like to note, that the two conceptual options outlined in this chapter do not rule out each other, but would also be possible to be implemented in parallel!

3.2. Interoperability updates of current ape profiles

Archival standards are available to encode archival descriptions (EAD), archival contexts in terms of corporate bodies, persons and families (EAC-CPF) and information about archives themselves (EAG). All three international archival standards have been adapted by the Archives Portal Europe and defined as ape profiles¹⁰.

Whereas the EAD and EAC-CPF standards are developed and supervised by an international technical sub-committee (maintained by the Society of American Archivists), the EAG standard has been developed by a project of Spanish and Latin American archives and revised by the APE_x project and CENDARI with mainly European partners. Nevertheless all three standards are more or less synchronised in structure and syntax.

As of now the following elements are already controlled by some syntactic or semantic rules.

Table 1: Elements currently controlled in ape standards

apeEAD elements / attributes	Default content / values
//language @langcode @scriptcode	Language of the description or the material encoded according to ISO 639-2b (three letters, international) Script of the description or the material encoded according to ISO 15924 (four letters)

⁸ See CIDOC Conceptual Reference Model at <http://www.cidoc-crm.org/> (viewed 2 October 2014).

⁹ See Europeana Data Model at <http://pro.europeana.eu/edm-documentation> (viewed 2 October 2014).

¹⁰ <http://apex-project.eu/index.php/outcomes/standards> (viewed 2 October 2014).

//eadid @countrycode	Country, in which the materials being described are held, according to ISO 3166-1 (two capital letters)
//c @level	Use the LEVEL attribute to identify the descriptive character of the component. apeEAD default values: "fonds", "series", "subseries", "file", "item"
apeEAC-CPF v0.2 elements / attributes	Default content / values
@xml:lang	Language of a single element's content encoded according to ISO 639-2b (three letters, international)
//script @scriptCode	Script of the description or used by the entity described encoded according to ISO 15924 (four letters)
//language @languageCode	Language of the description or used by the entity described encoded according to ISO 639-2b
@countryCode	Country according to ISO 3166-1 (two capital letters)
//nameEntry @localType	"authorized", "alternative", "preferred", "abbreviation", "other"
//nameEntry/part @localType	"corpname", "famname", "persname", "surname", "firstname", "birthname", "title", "prefix", "suffix", "alias", "patronymic", "legalform"
//nameEntryParallel @localType	"authorized", "alternative", "preferred", "abbreviation", "other"
//date @localType	"unknown", "unknownStart", "unknownEnd", "open"
//dateRange @localType	"unknown", "unknownStart", "unknownEnd", "open"
//address @localType	"visitors address", "postal address", "other"
//addressline @localType	"firstdem", "secondem", "postalcode", "localentity", "street", "country", "other"
//placeEntry @localType	"birth", "foundation", "private-residence", "business-residence", "death", "suppression", "other"
//relationEntry @localType	"title", "id", "agencyCode", "agencyName"
//cpfRelation @cpfRelationType	"identity", "hierarchical", "hierarchical-parent", "hierarchical-child", "temporal", "temporal-earlier", "temporal-later", "family", "associative"
//resourceRelation @resourceRelationType	"creatorOf", "subjectOf", "other"
EAG 2012 elements /	Default content / values

attributes	
@xml:lang	Language of a single element’s content encoded according to ISO 639-2b (three letters, international)
//script @scriptCode	Script of the description encoded according to ISO 15924 (four letters)
//language @languageCode	Language of the description encoded according to ISO 639-2b
//placeEntry @countryCode //repositorid @countrycode	Country according to ISO 3166-1 (two capital letters)
//location @localType	"visitors address", "postal address"
//num @unit	"squaremetre", "linearmetre", "site", "book", "title"
//resourceRelation @resourceRelationType	"creatorOf", "subjectOf", "other"
//eagRelation @eagRelationType	"hierarchical-child", "hierarchical-parent", "temporal-earlier", "temporal-later", "associative"

In addition to the specific elements all three standards (EAD, EAC-CPF, EAG) provide possibilities to encode information in two main areas:

- Administrative information (ead:eadheader, eac-cpf:control, eag:control) and
- Descriptive information (ead:archdesc, eac-cpf:cpfDescription, eag:archguide).

These elements can be used to provide a high level of referencing to the semantic assets used, however these elements are used on top of the whole description and are not explicitly associated to the specific elements where the asset has been used.

When looking at the single element level we can see that for now the support for semantic descriptions and referencing is mainly available in EAC. However, the ongoing EAD3 revision is also introducing some additional control possibilities and there are also some possibilities in the current EAD2002 which have not been carried over to apeEAD. Therefore it would make a lot sense to the Archives Portal Europe to keep a close eye on the efforts with EAD3 and take up suitable elements and methodologies in a possible revision of apeEAD.

Table 2: Elements and attributes supporting semantic references

apeEAD elements / attributes	Meaning
//corpname @authfilenumber	Number that identifies a correspondent authority file for a corporate body’s name; may be used as a link to the vocabulary item
//name @authfilenumber	Number that identifies a correspondent authority file for a (generic) name; may be used as a link to the vocabulary item

//persname @authfilenumber	Number that identifies a correspondent authority file for a person's name; may be used as a link to the vocabulary item
//famname @authfilenumber	Number that identifies a correspondent authority file for a family's name; may be used as a link to the vocabulary item
apeEAC-CPF v0.2 elements / attributes	Meaning
//localControl/term	Explicit element for linking to local, regional or national vocabularies
//conventionDeclaration/abbreviation	A declaration of the rules or conventions, including authorised controlled vocabularies and thesauri, applied in creating the EAC-CPF instance
//conventionDeclaration/citation @xlink:href	Declare references in the <citation> element and link with @xlink:href
//placeEntry @vocabularySource	Places should be identified by the proper noun that commonly designates the place, natural feature, or political jurisdiction. It is recommended that place names be taken from authorised vocabularies. @vocabularySource attribute may be used to indicate the controlled vocabulary from which the <placeEntry> term is derived.
//legalStatus/term @vocabularySource	The legal status of a corporate body is typically defined and granted by authorities or through authorised agencies. Enter terms in accordance with provisions of the controlling legislation. Terms may be drawn from controlled vocabularies or may be natural language terms.
//function/term @vocabularySource	Terms are used to identify the functions, processes, activities, tasks, or transactions performed by the CPF entity described in the EAC-CPF instance. They may be drawn from controlled vocabularies or may be natural language terms.
//occupation/term @vocabularySource	Terms are used to identify an occupation held by the CPF entity. Terms may be drawn from controlled vocabularies or may be natural language terms.
EAG 2012 elements / attributes	
@source attribute next to elements autform, parform, nonpreform,	The source attribute allows to refer explicitly to the vocabulary from which the appropriate nameform of the

repositoryName	institution (authorised, parallel, nonpreferred) has been derived from
//conventionDeclaration/abbreviation	A declaration of the rules or conventions, including authorised controlled vocabularies and thesauri, applied in creating the EAG instance
//conventionDeclaration/citation @xlink:href	Declare references in the <citation> element and link with @xlink:href
General EAD2002 elements / attributes	Meaning
//corpname @source	For identifying the names of organisations or groups of people that act as an organisational entity and are related to the materials being described. Use the attribute @source to name the authority repository, where the controlled term comes from.
//famname @source	For identifying a group of persons closely related by blood or persons, who form a household, and are related to the materials being described. Use the attribute @source to name the authority repository, where the controlled term comes from.
//name @source	The proper noun or noun phrase designation for an entity that is difficult to tag more specifically. Use the attribute @source to name the authority repository, where the controlled term comes from.
//persname @source	For identifying a person, including any or all of that person's forenames, surnames, honorific titles, and added names, who is related to the materials being described as either a source, creator, or subject. Use the attribute @source to name the authority repository, where the controlled term comes from.
//function @source @authfilenumber	For specifying activities and processes that generated the described materials. Use the attribute @source to name the authority repository, where the controlled term comes from and use the attribute @authfilenumber to link to the authority file.
//genreform @source @authfilenumber	For identifying the types of material being described in controlled access headings or a structured statement of physical description, by naming the style or technique of their intellectual content (genre); order of information or object function (form); and physical characteristics. Use the attribute @source to name the authority repository, where the controlled term comes from and

	use the attribute @authfilenumber to link to the authority file.
//geogname @source @authfilenumber	For indicating a place, natural feature, or political jurisdiction. Use the attribute @source to name the authority repository, where the controlled term comes from and use the attribute @authfilenumber to link to the authority file.
//occupation @source @authfilenumber	For specifying a type of work, profession, trade, business, or avocation significantly reflected in the materials being described. Use the attribute @source to name the authority repository, where the controlled term comes from and use the attribute @authfilenumber to link to the authority file.
//subject @source @authfilenumber	For indicating a topic reflected in the described materials. Use the attribute @source to name the authority repository, where the controlled term comes from and use the attribute @authfilenumber to link to the authority file.
//title @source @authfilenumber	The formal name of an intellectual work, such as a monograph, serial, or painting, listed in a finding aid. Use the attribute @source to name the authority repository, where the controlled term comes from and use the attribute @authfilenumber to link to the authority file.
EAD3 elements / attributes	Meaning
//unittype @source @identifier	Within a structured statement of physical description, indicates the nature of the unit being quantified. Use the attribute @source to name the authority repository, where the controlled term comes from and use the attribute @identifier to link to the authority file. [Note, that @identifier will generally replace @authfilenumber in the revised version of EAD.]

As a summary we can say that while there are already now some possibilities for semantic referencing, though not yet used in apeEAD to their full extent, then additional work should be done to develop a common approach either by more actively using the given possibilities in EAD (attributes for *source* and *authfilenumber* [EAD 2002] respectively *identifier* [EAD3]) or taking up methods from the RDF and LOD areas.

Partly related to this work is also our recommendation to start working on an exhaustive domain model of the archives, record creators, fonds, object types, etc., together with the whole range of possible content in the objects (persons, locations, abstract concepts, historical events, etc.). Such a model would make it much easier to identify the necessary elements for standardisation as well as link similar elements in different standards¹¹. In addition it would allow us to create tools to:

- Translate existing metadata (both about the objects and their content, for instance the //ead:controlaccess child elements) into other languages;
- Create new metadata based on the ontology that makes the content more accessible by enabling semantic searches regardless of language or accessibility of the actual content.

However, we can see already now that there are some elements in apeEAD, which we recommend to start using in a controlled way in the Archives Portal Europe as soon as possible.

Table 3: Elements recommended for central semantic control

apeEAD elements / attributes	Default content / values
//physfacet @type	The TYPE attribute may be used to specify which aspect of the physical appearance is being designated. apeEAD has no default values here.
//unitid @type	The TYPE attribute may be used to indicate the system from which the <unitid> was derived. ApeEAD has no default values defined here in the schema, though effectively there are certain values used in the data processing to differentiate between a current “call number”, a “former call number” and a “file reference” originating from the records creator’s own filing system.

3.3. Centrally recommended semantic assets in the Archives Portal Europe

Regardless of the support level of semantic assets (ie built-in or referenced) there are some specific ontologies which should be looked upon in more detail as they provide most potential for widening the access possibilities and are, at the same time, most difficult to implement. These ontologies are:

- Authority and person names;
- Place names;
- Content subjects / keywords;
- Time periods;

¹¹ Please also note that the International Council on Archives has established an Expert Group on Archival Description (EGAD) which is expected to work until 2016 on similar issues. Though little information about the progress and exact actions of this group was available by the time of compiling this deliverable the Archives Portal Europe will try to monitor actions and reuse any feasible outcomes as soon as these emerge.

Authority names: Especially for hobby historians (genealogists) the search for persons and other names, which are semantically “tagged” in the archival descriptions, can be relevant. Thus having an authority name ontology might provide great benefits in both faceted search and also linking of archival records.

Creating a comprehensive ontology about person names is of course impossible. However, there might be possibilities to start with that in a limited manner by introducing first only the entities, which are mentioned in the EAC-CPF files gathered in the portal.

The **ontology structure** could be based on some of the widely used ones like Friend-of-a-friend (FOAF, <http://www.foaf-project.org/>, viewed 2 October 2014). However, it seems currently that it would make much sense for the Archives Portal Europe to start the actual ontology as a specific Archives Portal Europe authority ontology. The reasons for this recommendation are that the relations and structures between persons and agencies related to archival records and descriptions are rather specific and according to our current knowledge and paper research no suitable ontology exists to describe these, in addition the Archives Portal Europe is slowly emerging as the main international repository for authority files. However, this ontology could be based upon and/or mapped to:

- The Library of Congress name authority file (<http://id.loc.gov/authorities/names.html>, viewed 2 October 2014);
- The Virtual International Authority File (<http://viaf.org/>, viewed 2 October 2014);
- DBPedia (www.dbpedia.org, viewed 2 October 2014, also used by Europeana).

However, for the entities, which do not act as archival creators, it still remains to be discussed whether it might be reasonable to simply use something like DBPedia or extend the Archives Portal Europe authority ontology to include additional information.

Place / location: Another type of ontology, which potentially is highly interesting for the users of the Archives Portal Europe, is the place / location name.

As of now some place names are already encoded in ead//controlaccess/geogname, also EAC-CPF provides some tools for place codification. Therefore the mapping of a place name ontology to archival descriptions could be done similarly to the approach for person names (ie start with the explicit place name elements in EAD/EAC-CPF, continue with automated mapping by some tools and double-check these by simple tools in the Dashboard or Portal).

However, there are some problems, which need to be solved in addition:

- A place in a description might have different roles. As an example a named birthplace of a person is not necessarily related to a specific manuscript, an agency, whose location is mentioned, might be the original creator or the current holder etc. Therefore it needs to be looked upon how to extract and describe the relations of entities to archival records / description units;
- As place names differ in different languages also the support for multilingualism needs to be added. Ie - the ontology applied in the Archives Portal Europe needs to support the information of “Köln is the same as Cologne”.

For place names there are already multiple ontologies available outside the archival world and it seems not reasonable to create a totally new one. Our recommendation would be to use

GeoNames (<http://geonames.org/>), which is also used by Europeana and is in general currently the leading ontology in this field.

Subject and keywords: Also a central ontology of subjects / keywords seems to be highly reasonable. However, creating such a list of controlled subject headings is probably also one of the most difficult items to be implemented for Archives Portal Europe. Namely, it is almost impossible to use automated tools to add controlled subjects to descriptions of archival records. Also in situations where subjects are already available inside descriptions these are based on a variety of languages and different vocabularies, which are time-consuming to be mapped with each other.

Still, some partial solutions might be possible. Especially the possibility of exploiting higher level descriptions (ie describing subjects mainly on fonds / series level) could be considered for tagging archival descriptions. For already available subject headings also partial mapping of only most relevant terms into a central listing is possible. Still, even these partial solutions demand a high level of manual work and the quality and relevance of search results might still not be sufficient for the user.

We can also identify some of the subject ontologies which could be reused by the Archives Portal Europe or provide the main building blocks for a new ontology:

- DDC (Dewey Decimal Classification, http://en.wikipedia.org/wiki/Dewey_Decimal_Classification, example in applying DDC as LOD - <http://dewey.info/class/641/about>, both links viewed 2 October 2014);
- UKAT (UK Archival Thesaurus, <http://www.ukat.org.uk/>, linked data service available at <http://data.aim25.ac.uk/>, both links viewed 2 October 2014);
- LCSH (Library of Congress Subject Headings, <http://id.loc.gov/authorities/subjects.html>, viewed 2 October 2014);
- To map better to Europeana the possibility of the GEMET concept thesaurus could also be researched (<http://www.eionet.europa.eu/gemet>, viewed 2 October 2014).

As such our recommendation for the Archives Portal Europe is to investigate the mentioned semantic assets, select the most appropriate one and recommend data providers to implement it, thus also providing relevant guidelines. However, we should not expect that a reasonable amount of controlled values would be available in the near future and therefore also the use of this ontology within search functions should be limited.

Time periods: Additionally to standardised date formats, which are used for result refinements, the definition of time periods, valid for the European countries, would be intuitive and user-friendly. For standardised dates, the Archives Portal Europe data have a quite good quality, as this information is generated automatically during the conversion process. The according attributes in apeEAD, apeEAC-CPF or EAG need to be retrieved and assigned to the defined period.

Defining time periods valid for all European cultures and archives is a challenging task, as time slots for a period might change between the countries. As noticed in D6.1, a selection of historical, cultural or social epochs could be defined and related to certain dates and periods. Another possibility is also to take into account the different durations of epochs in different regions. This would mean that the APE_x project defines centrally general cultural periods like ancient history, middle age, modern age, and contemporary history. Each country can specify the exact time span for each epoch regarding the specific circumstances in this country.

A third approach is the usage of Semium Time¹², a vocabulary of time periods originally developed for Europeana. Semium Time is an ontology available as RDF/SKOS and under CC BY-SA licence. As soon as the APE_x project or the Archives Portal Europe foundation decides go for Linked Data or to enhance archival descriptions automatically, Semium Time could probably be used relatively easily in order to provide user-friendly date information.

3.4. Tools and methods to support semantic assets in the Archives Portal Europe

This chapter describes some of the possibilities, tools and methods, which might be used and/or integrated into the portal to support the management of semantic assets.

a) Uploading semantic assets

If the Archives Portal Europe develops a generic environment to manage the semantic assets, which have been used in archival descriptions then the first question is which format to use for the assets.

As of now RDFS or OWL seem to be the most appropriate languages for this. A wide range of tools supports both and it would be possible to add abilities similar to the validation and conversion of XML files to the Data Preparation Tool and the Dashboard, in which such tools should be integrated to provide a unified user experience for data owners.

b) Managing semantic assets

Building a whole new environment from scratch is not reasonable and therefore we have done some paper research on already available tools, which might possibly be used as the backbone. The list of some of these tools, which could be evaluated further in future, is as follows:

- Ontotext AD is a developer of core semantic technology, text mining and web mining solutions. It is also used by Europeana. (<http://www.ontotext.com>, viewed 2 October 2014)
- KAON Ontology Framework: “KAON is an ontology management infrastructure targeted for business applications. It includes a comprehensive tool suite allowing easy ontology creation and management. Persistence mechanisms of KAON are based on relational databases.” (<http://sourceforge.net/projects/kaon/>, viewed 2 October 2014)
- Protégé: “Protégé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. At its core, Protégé implements a rich set of knowledge-modelling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Protégé can be customized to provide domain-friendly support for creating knowledge models and entering data. Further, Protégé can be extended by way of a [plug-in architecture and a Java-based Application Programming Interface \(API\)](#) for building knowledge-based tools and applications.” (<http://protege.stanford.edu>, viewed 2 October 2014)
- SAS® Ontology Management: “SAS Ontology Management defines and manages semantic terms that can be used to organize and process content from different systems, silos and

¹² <http://semium.org/time.html> (viewed 2 October 2014).

repositories across your organization. By creating ontologies that are centrally managed, you can examine content through a sophisticated, informed lens – maximizing content value.” (<http://www.sas.com/text-analytics/ontology-management/>, viewed 2 October 2014)

- Smartlogic Ontology Manager: “Advanced Ontology Software for Building, Enhancing and Browsing Semantic Models. Semaphore Ontology Manager is at the heart of the Semaphore Enterprise Semantic Platform solution. This software covers the life-cycle of taxonomy, thesauri or ontology development and maintenance.” (<http://www.smartlogic.com/home/products/semaphore-modules/ontology-manager/ontology-manager-overview>, viewed 2 October 2014)
- Synaptica Enterprise Taxonomy Management Software: “Synaptica Enterprise is our behind-the-firewall solution for larger organizations. Enterprise systems are available with either perpetual or subscription licences. They are designed to support multiple editors with role-based permissions as well as compartmentalization and collaboration workspaces. They also provide access to a suite of integration tools, including database APIs and Web Services. Enterprise systems start at \$25,000 and can be scaled up from single to unlimited taxonomy systems as needed.” (<http://www.synaptica.com/products.html>, viewed 2 October 2014)
- Thesaurus Master™: “Thesaurus Master® is a cutting edge software tool for taxonomy and metadata management. A controlled vocabulary is absolutely necessary if you want precise document indexing and accurate information retrieval. Whether you develop this vocabulary (term list, authority file, thesaurus) yourself or obtain it whole or in part from external sources, Data Harmony®’s Thesaurus Master puts you in control of the entries, the hierarchy, and the conditions of use.” (http://www.dataharmony.com/products/thesaurus_master.html, viewed 2 October 2014)
- Wordmap Taxonomy Management System: “Wordmap Taxonomy Management System ("Wordmap") is used by information professionals to develop the category and term sets that bring consistency, precision and control to enterprise information management. Wordmap's broad feature set gives editors much greater control and productivity than they would enjoy in other applications. Yet Wordmap is outstandingly easy to set up and use. The data and functions of Wordmap are available via imported and exported XML, a documented Java API and taxonomy connectors. Wordmap can thus provide enterprise-class integration capabilities.” (<http://www.wordmap.com/what-is-wordmap>, viewed 2 October 2014)

3.5. Use cases for managing semantic assets¹³

Each of the use cases below follows a simple structure:

- *Rationale*: short description of the use case;

¹³ Please note that the use cases here and the chapters below have also been used for the deliverable D4.7/D6.8.

- *Use case*: longer description of the use case, some with highlighted questions to be answered.

UC1. Uploading semantic assets Into Archives Portal Europe Dashboard

Rationale	Following the files containing archival descriptions (apeEAD), those containing information about the creators of the archival materials (apeEAC-CPF) and the ones containing information about the archival institutions (EAG 2012), the content providers also upload the semantic assets which they have used in these three data types.
Use case	<p>The content provider logs in the dashboard and selects the section “semantic assets”. In this section the content provider selects “upload new assets”. By doing so the content provider gets an input screen where the name, short description, type, format and web links can be entered. Assets can also be uploaded directly in this section.</p> <p>The content provider can also choose whether the asset should become available in the archivesportaleurope.net namespace or the original namespace be kept.</p> <p>By clicking “next” the dashboard runs a few automated checks on the quality, validity etc. of the semantic asset. If errors occur, they will be listed on the screen, otherwise the semantic asset appears on the dashboard and is visible to all users.</p>

UC2. Training¹⁴ a semantic engine to use the semantic assets

Rationale	The administrator and/or semantics expert is allowed to train a semantic engine used inside the Archives Portal Europe to support semantic assets available on the dashboard.
Use case	<p>The administrator logs in the dashboard where a section of “semantic tools” is available. (S)he opens the section and sees a list of tools and ontologies supported inside Archives Portal Europe.</p> <p>Next to it (s)he can see the list of newly added semantic assets and notices that a new high quality asset has been added recently. (S)he finds an appropriate example data, adds the ontology to the tool, and starts the training process. After the training process has been finished, the support for the ontology inside the tool is published on the dashboard.</p> <p>Note: The “administrator” is used here as a common term for an IT specialist who has special access rights inside the dashboard to configure the semantic tools. Most</p>

¹⁴ By default semantic engines like Named Entity Recognition Tools and Natural Language Processing toolkits support a limited set of languages and vocabularies. However, some toolkits also include “training” functionality, which allow the analysis and learning of prepared corpora of common texts which have been manually prepared. Therefore it is possible to introduce new languages and vocabularies to the tools with training capability without the need of re-programming.

	probably such person needs to be available from multiple countries.
--	---

UC3. Linking archival descriptions to semantic assets

Rationale	A content provider is allowed to use tools to link archival descriptions to available semantic assets.
Use case	<p>The content provider goes to the dashboard and selects the content manager section. In the content manager the content provider is allowed to select one or many finding aids, holdings guides, source guides, and/or authority records and apply the task “semantic annotation”.</p> <p>When doing so the content provider gets a dialogue box that asks, which entities are to be annotated and the exact assets to be used. By default the recommended ontologies of the Archives Portal Europe are selected, but the content provider can also choose other ontologies supported by the semantic tools in the Archives Portal Europe.</p> <p>After selecting the tool and the ontology the enrichment process starts. The user is then prompted with a summary of all the results. Based on this summary the content provider is able to start a manual review of the results. In addition the content provider has the possibility to create a crowdsourcing task to proof read the automatic results where the users would be allowed to carry out the same task.</p> <p>Note: This use case overlaps partly with UC5 on Named Entity Recognition.</p>

UC4. Managing semantic assets

Rationale	The content provider is allowed to carry out some simple management of the semantic assets, which have been uploaded to the dashboard.
Use case	<p>The content provider goes to the dashboard and selects the semantic assets section. There (s)he can see a list of available assets. Next to each of these a set of management actions, among which are the possibility to connect it to other semantic assets, organise and amend it, export in specific formats etc.</p> <p>Note: This Use Case is a placeholder for further discussions in the future.</p>

4. Named Entity Recognition

One of the most crucial technologies behind both LOD and tagging is the Named Entity Recognition, which is a part of Natural Language Processing tools.

Simply put Named Entity Recognition (NER) tools help you to automatically identify which parts of your data might match a value in an ontology or vocabulary, thus helping you to annotate your data much more quickly.

4.1. Evaluating Named Entity Recognition tools

We started the work around NER tools by putting together a list of tools to be evaluated. Most of the tools were derived from the appropriate Wikipedia article¹⁵ of Natural Language Processing tools and a first selection was done to find the ones with NER capabilities. In addition some tools not available in the first list were added based on participants' previous knowledge and further desktop research.

The final list of tools included 33 software solutions for which we assembled the following information:

- Name of the tool;
- Homepage;
- Developer;
- How widely is it used;
- Support for multiple languages;
- Is the tool open source or not;
- Has it been implemented in memory institutions;
- Support for semantic assets and extensibility;
- Programming language used.

The full list with the information we were able to find from the web is available in Annex I.

Using this information the next step was to select the tools, which seemed most promising. The main criteria for the selection were the support for languages, open source installation possibility and extensibility. As secondary criteria also the programming language and use in other memory institutions was taken into account.

In general we can say that there are no tools available, which would allow reasonable support for most European languages. The majority of available tools support only a single language and a few ontologies. At the same time there are some tools, which do not only implement specific ontologies but also allow using your own ones, as such acting more as a NER framework than a specific tool.

As the result the following five tools were selected for closer testing and hands-on evaluation:

¹⁵ http://en.wikipedia.org/wiki/List_of_natural_language_processing_toolkits

- GATE;
- Stanford NLP;
- UIMA;
- NERD;
- YAGO.

For all of these tools a more comprehensive evaluation report has been created which is also available in Annex II.

Based on the hands-on evaluation we recommend the technical team of the Archives Portal Europe to consider:

- Adjusting and implementing the Apache UIMA framework¹⁶. It is currently one of the most widely used frameworks which is at the same time offering a range of possibilities and yet rather simple to use and configure. Especially interesting is the possibility to implement it using the ontologies defined or taken up by the Archives Portal Europe and rather simply to connect to the ontology management to add additional links between ontology values in multiple languages;
- Implementing NERD. It is according to our tests the best “preconfigured” tool available. While it is not possible to use it easily with custom ontologies it acts as an umbrella for 12 different tools, including AlchemyAPI¹⁷ and DBpedia Spotlight, therefore bringing support to multiple ontologies in one package. The implementation of NERD could be the fast and simple alternative before UIMA has been set up to deal with more specific and multilingual ontologies inside the portal.

When looking at Stanford NLP then this solution is rather comparable to the UIMA framework. However, in our opinion it is harder to implement and customise and therefore UIMA should be preferred.

GATE is rather simple to implement initially but, while allowing for some customisation, adding new ontologies is not as simple as for UIMA. At the same time the preconfigured settings provide not as good results as NERD.

YAGO is an interesting conceptual solution but according to our information it has not been in active development recently. As such it cannot be recommended due to doubts around sustainability.

During spring 2014 the APE_x project has also partnered with the CENDARI project¹⁸, which is currently developing a multilingual NER tool for its purpose of annotating historic documents based on their ontologies. According to current planning the tool will be available by the end of 2014 and in this case it could also be evaluated and possibly recommended to be included into the Archives Portal Europe.

¹⁶ <https://uima.apache.org/> (viewed 2 October 2014).

¹⁷ <http://www.alchemyapi.com/> (viewed 2 October 2014).

¹⁸ <http://www.cendari.eu/> (viewed 2 October 2014).

We also recommend promoting the use of Open Refine¹⁹ for entity recognition and linking. It is probably the most powerful tool to be used but on the downside requires some time to learn before starting to use it. However, some good “How To” guides are available for this purpose (as an example as part of the LOD Handbook²⁰ produced by the YEAH! project, a good tutorial is also available from the Technical University of Delft²¹).

Please also note that the work on NER capability inside the Archives Portal Europe will continue with the aim to produce more specific technical guidelines on integrating UIMA and NERD into the Dashboard.

4.2. Use cases for Named Entity Recognition tools

UC5. Using Named Entity Recognition on Archives Portal Europe

Rationale	The content provider makes use of the Named Entity Recognition tool(s) to analyse their data.
Use case	<p>The content provider enters the content manager section inside the dashboard and selects one or more finding aids. (S)he then selects the task “Entity Recognition” and executes it. In case multiple tools are available, the system automatically recommends the most appropriate one (based on language and other settings).</p> <p>As a result the content provider gets a list of contextual hits, including the full sentence where the entities were found, the entities themselves and their types (ie place names, person names, etc).</p> <p>S(he) also gets a summary report, highlighting how many hits were found in total and how many of these are person names, place names, dates, etc.</p> <p>(S)he is allowed to select whether (s)he approves the results as part of the descriptions, or keep them as her/his “recommendations”. In case of “recommendations”, the entities need to be later approved by other content providers, or through the means of crowdsourcing, before they are published and used within the Archives Portal Europe.</p> <p>Note: This use case overlaps partly with UC3 above, though here we do not assume that a direct link to a semantic asset is recorded.</p>

¹⁹ <http://openrefine.org/> (viewed 2 October 2014).

²⁰ https://pure.ltu.se/portal/files/96261604/YEAH_Handbook_ver_1_1_20140506.pdf (viewed 2 October 2014).

²¹ http://enipedia.tudelft.nl/wiki/OpenRefine_Tutorial (viewed 2 October 2014).

5. Tagging

5.1. Relation to deliverable D6.1

As already described in Deliverable D6.1²² the Archives Portal Europe looks at tagging mainly from the user perspective, therefore we are mainly talking about tagging as a crowdsourcing activity where users can add new tags or – in relation to the NER capability described in the previous chapter – confirm predefined tags.

The previous deliverable D6.1 already covers a lot of ground in the discussions around user tagging. Therefore we highly recommend reading its chapter 10 before continuing. In short, the chapter concentrates mainly on the major problems and research questions related to user tagging and documents discussions around the scope of user tagging, quality control and publishing, use of controlled vocabularies, reuse of user generated tags and motivating users. As well, some good examples are provided from other memory institutions and a component architecture is proposed.

The purpose of this chapter is therefore to rather give a short outline how user tagging could be exploited in relation to creating more reasonable LOD and not repeat the previous discussions.

5.2. Combining user tagging and Named Entity Recognition

When looking at the current capability of NER tools we can see that these are not able to provide 100% accuracy. Much information is available on the web on the precision of these tools but in general we could say that most usually the accuracy remains between 80 – 90%. As well, this depends very much on the language used (mostly results are better for English than for other languages), whether the NER tool has been specially trained to handle the texts to be analysed, which ontologies have been used etc.

Therefore one of the crucial tasks to be done is to do some sort of quality approval on the results acquired automatically by the NER tools. One of the possibilities is to mix the automated task of the Named Entity Recognition with the power of crowdsourcing and let the crowd approve the results. Simply put – when the NER tools output automatically some named entities and state something about these (as an example that “Berlin” is a “city”) we can give this information to the users and let them only state a simple “yes” or “no” to approve this.

We can see that such an approach would allow us at the same time to:

- Speed up the tagging exercise by automated means;
- Keep crowdsourcing activities very simple and efficient;
- Therefore provide much more quality on more content than would be possible when using only NER or user-tagging alone.

Of course, there are also some dangers to it. Especially we can get to a situation where data in different languages is handled unequally, meaning that as we probably get much more entities

²² http://www.apex-project.eu/images/docs/D61_Web20_In_Archival_Applications.pdf (viewed 2 October 2014).

recognised in English than in other languages then also the majority of crowdsourcing tags and “confirmed entities” would be on top of English data which at the same time forms only a small part of the descriptions on the Archives Portal Europe.

As well, we need to ensure that the crowdsourcing task does also provide some level on context to the recognised entity. It would be most straightforward to display the whole sentence to the users. As an example only having sentences like “Biography of Irving Berlin” or “Berlin City Hall construction plans” would allow users to decide whether the recognised entity “Berlin” is indeed a place name or not.

As a summary we highly recommend the Archives Portal Europe to apply a combination of NER and user-tagging on top of the data. However, especially the selection of the NER tools needs to be looked upon carefully; with an addition to specifically investigate possibilities for training the Apache UIMA framework to support archival descriptions of multiple languages beforehand.

5.3. Reusing user-generated tags for Linked Data

The tags, which have been automatically identified and then confirmed by users, can be used for multiple purposes, most notably by faceted queries and for the provision of Linked Data.

During the discussions around the reuse of these tags we have come across two major issues, which need to be dealt with:

- Quality and completeness: most probably institutions would not be willing to publish the tags in their data unless they are either reasonably satisfied with the quality and completeness of these. However, we can also see that the definition of “sufficient quality and completeness” can vary a lot and therefore we recommend allowing for some flexibility in the according functionality. As the rule of thumb we recommend not to include the user confirmed tags into the data by default but to allow institutions to state whether they:
 - Do not allow user tagging at all;
 - Want to review all user tags manually;
 - Approve all user tags by default;
 - All user tags are highlighted on the portal but separately from the official description in a “descriptions provided by users” section;
 - etc.
- Entities to be used: while LOD could “link” a variety of different entities we recommend to keep at least initially the scope of tagging and therefore also the use of “linked entities” as a minimum. Our recommendation is to initially only look upon the entities of place and authority names, possibly also dates and keywords. This recommendation overlaps also with the ones made in chapter 3 above as well as the scope of LOD entities in Europeana²³.

5.4. Use cases for user-tagging

²³ For more information see <http://labs.europeana.eu/api/linked-open-data/data-structure/>

UC6. User-tagging by approving automated tag matches (archivist view)

Rationale	The archivist prepares a crowdsourcing task based on the recommendations found in UC5
Use case	<p>The archivist has previously run NER on her/his data and has now a set of recommendations, which need to be approved.</p> <p>The archivist goes to the Dashboard to the Content Manager section and selects the finding aids, which have open recommendations. Next the archivist has the possibility to select “define crowdsourcing task”, which opens a form prefilled with the information about the finding aid(s) – number of open recommendations, their type, language etc., all very similar to the NER report.</p> <p>Next the archivist is able to select whether all or only part of the recommendations become part of the crowdsourcing task (as an example, only place names can be selected). As well the archivist can select whether users need to be logged in or not and how many users need to approve the recommendation (1e – two, three or four overlapping results for a recommendation need to be available to confirm the recommendation).</p> <p>Next time the archivist logs in (s)he can see the progress of the crowdsourcing task – how many recommendations have been approved, how many have contradictory responses, how many have not sufficient responses. Based on this information the archivist can decide to keep the task open or finish it.</p> <p>When finishing the crowdsourcing task the archivist can confirm that all approved recommendation become part of the official descriptions and review all open or contradictory responses.</p> <p>Note: As an alternative the setup of crowdsourcing tasks could also be a separate section on the Dashboard</p>

UC7. User-tagging by approving automated tag matches (user view)

Rationale	The user executes a crowdsourcing task on the portal.
Use case	<p>The user enters in the Archives Portal Europe and browses to the “Contribute to our portal” section. Amongst other functions, (s)he finds the “tagging missions” functionality, through which the user can start confirming automatically found recommendations.</p> <p>Users can see a list of “tagging missions” with some additional information – how many tags still need to be approved, what language is used for the tags, what type of tags there are (place, person names, dates and time, keywords), and also from what archives the content derives.</p> <p>Users can either choose to select one of the collections, or go for the “random” mode, which presents tags from all tagging missions.</p>

	<p>When executing a tagging mission, the user is directed to continuous screens, where (s)he can state whether a tag recommendation is correct or not. There are also options for skipping a recommendation, or exiting the crowdsourcing mission.</p> <p>Note: The question whether a user needs to log in or not is handled in this use case, but nevertheless, it needs to be discussed and agreed upon at some point.</p> <p>It also has to be discussed further, if and how the tagged information could be delivered to the original content providers, so that they could reuse it.</p>
--	---

UC8. “Open” user-tagging

Rationale	The users are allowed to approve tags without going into any tagging missions.
Use case	<p>When a user browses the archival descriptions, (s)he can also see recognised entities, in different fonts and underlined. When the user mouses-over such entities, (s)he is allowed to see some information about what type of entity it is, and make a simple query for all information which has been tagged with this entity.</p> <p>In addition, users have the possibility to approve or reject an entity type if it is displayed incorrectly, or also to select random words in archival descriptions and indicate the entities of a certain type.</p> <p>As an example: the user browses through the portal and discovers that within an archival description, the word “Berlin” is marked, as being automatically defined, as a “name of person”, though in this context being related to the place of that name. With a single click the user can report that the automated recognition is “not correct”.</p> <p>Note: Same as for the previous use case, the question whether a user needs to be logged in or not is not addressed in this use case but nevertheless needs to be discussed and agreed upon at some point.</p>

6. Persistent Identifiers

6.1. Rationale

In the course of discussion, Persistent Identifiers (PID) and the infrastructure to support these have been always an issue for various tasks in the Archives Portal Europe, but the discussion has been postponed several times due to the priorities of other tasks and the difficulty of the issue.

Identification as such and also the persistency of identifiers **must be** tackled and practical outcomes need to be available by the end of the APE_x project in early 2015. Essentially a large set of functionality - including featured documents, bookmarking, LOD and any other kind of external referencing - is in desperate need of a persistent identification model which would allow users and data owners to be sure that links created to objects will remain valid as long as possible and are not “broken” due to system updates or any other technical issues.

Therefore the application of PID is a central matter for the sustainability of the portal. It has to be the mission of the APE_x project and the Archives Portal Europe to minimise the risk of broken links and ensure that the content is accessible in the long term. We should regard this as one of the basic infrastructural components of the Archives Portal Europe, rather than just a piece of added value.

6.2. Crucial characteristics of identifiers

An identifier is, by definition, “a sequence of characters used to identify or refer to a program or an element, such as a variable or a set of data, within it”²⁴. In the case of the Archives Portal Europe the main data to be identified are the descriptive units, though there are some additional elements, which are discussed in chapter 6.4.2 below.

The rest of this chapter discusses the most crucial characteristics of an identifier ecosystem – uniqueness, human-readability, management, and persistence, from the viewpoint of the Archives Portal Europe.

Uniqueness: While the definition of an identifier does not mention uniqueness explicitly, we can implicitly deduce that each identifier **must be** unique within its environment of use (otherwise it would identify multiple objects and thus not serve its purpose).

Therefore it is important to discuss what the general environment of the Archives Portal Europe is and essentially make clear whether we need identifiers, which are either:

- Unique just within the Archives Portal Europe (ie uniqueness on system level);
- Unique in the cultural heritage domain (ie uniqueness on domain level) or
- Globally unique.

Looking at the core needs of the Archives Portal Europe (as highlighted in the use cases in chapter 6.5 we can see that in short term system level uniqueness might be satisfactory. It is possible to solve most of the issues of referencing by local solutions, which do not necessarily take into account identifier systems in other systems either in the cultural heritage domain or globally.

²⁴ Definition derived from the Oxford Dictionary of English 2010 release, ISBN 0199571120.

However, in long term we have to take into account that identifiers, which are only unique in Archives Portal Europe, might become an obstacle when linking and connecting similar data in different portals or locations. The practical situation nowadays is that archival institutions need and want in parallel to share their data locally from their own catalogues, as local LOD, from Archives Portal Europe, Europeana and numerous other more or less specific central portals. As such there is clearly a long-term need for at least domain level uniqueness or even for global uniqueness.

However, this question can technically be solved rather easily by assigning a distinct domain extension (ie namespace) to an internally unique identifier (ie using hierarchical identifiers).

Human readability of identifiers. Another question is whether IDs should be human-readable - ie identifiers which reveal something about the content or context of the identified object – or meaningless – identifiers which are provided purely by using the uniqueness of certain mathematical algorithms.

In the archival world there is a strong candidate for meaningful identifiers, which is the hierarchical archival reference number (as an example the reference number used in the National Archives of Estonia is similar to EAA.1.2.3-4.5 where the first part – EAA – identifies the repository and the latter part the hierarchy of descriptive units). These reference numbers have been used in archival institutions already throughout decades and also the users are used to these. The main benefit of the reference number is that it is fairly easy for archivists and users to say in which archives or repository the identified object resides, whether it is a high level descriptive unit or a single item etc. In case of the Archives Portal Europe it would be quite straightforward to take the available numbers and simply add the Archives Portal Europe namespace, country code and institution code; all of which are already available in the portal.

However, the problem with these reference numbers (and in more general, with most meaningful IDs) would be that the original hierarchy (ie the meaning) could change over time. As an example, when archival institutions merge or collections are being transferred from one archive to another then also the identifiers would change. As well, archival records within archives might be rearranged so that they get a new place in the description hierarchy and thus a new reference number.

While these problems are possible to be solved by using technological and administrative solutions (as an example, every archive ensures that old identifiers will still be maintained after new ones have been assigned and redirection is offered) the confusion which might arise (ie archivists think that a record is kept in a specific archives while it actually has been transferred to another) and the risk of collisions (the old identifier has been assigned to a new item after the old one has been rearranged) make the use of meaningful IDs less stable.

As a summary we can see that both meaningful and meaningless IDs have their positive and negative sides. While we tend slightly to prefer the meaningless IDs for central and persistent identification we do not rule out the use of meaningful IDs in case the risks mentioned above are dealt with and necessary systems in place.

Regardless of which system is going to be implemented we can see that there is the crucial need for maintaining also historic identifiers (ie the ones which are outdated and not actively used) at the data providers' sites and also inside the Archives Portal Europe. Luckily the possibility of including multiple IDs and describing their context and use is already available in the standards used in the

Archives Portal Europe (EAD, EAG, EAC) and it also is common practice in archives to keep track of changes in reference numbers.

Management and persistence of IDs. One of the most crucial questions with identification is about defining the risks, which endanger the persistence and uniqueness of IDs and establish organisational and technical management tasks to mitigate these risks.

The main risk related to the persistence of IDs is the migration of the portal to new software platforms. As the idea of the Archives Portal Europe is to continue as a central access portal for decades to come we can see that any identifiers used in the portal need to be platform independent so that when updating the portal (and most probably assigning new database-level IDs) the identifiers assigned before will still remain intact.

In the ideal case this risk is managed by deploying an independent “PID resolver” system on top of the portal, which maps the persistent IDs into database-level IDs and allows for some level of management (most notably carrying out bulk operations of updating the mappings in case the internal rules in the system have changed).

Some additional risks to persistence are also mentioned above and below (in sections for uniqueness and meaning of identifiers, as well in the next chapter “Scenarios for Creating identifiers”). Some possible use cases for managing the persistence follow also below in chapter 6.5 Use cases - using PID for Archives Portal Europe functionality.

6.3. Scenarios for creating identifiers

When assigning IDs we have to take into account that the characteristics mentioned in the previous chapter are well discussed and met. The next step after that discussion should be to determine which scenario and location is the most reasonable one to create the actual identifiers – whether it should be done by the data providers themselves, should the Archives Portal Europe assign these or should we use some third-party services for this purpose. Below we have highlighted some of the main considerations for each of these scenarios:

- Reusing IDs created by data providers: the creation of IDs by data providers themselves would be the wished-for solution for the Archives Portal Europe. However, this also means that some of the tasks around the management and ensuring persistence of IDs would be left for the data providers and as such demand some rework and updates to the systems in place. Especially for smaller archival institutions there might even not be any catalogue systems and if, it would be hard to come up with relevant funding in many cases. However, this might be a reasonable solution for larger institutions, which do provide most of the content for the Archives Portal Europe;
- Creating IDs inside the Archives Portal Europe: the creation of IDs by the Archives Portal Europe would be a reasonable solution in terms of cost (for both the tools to create IDs and also manage these afterwards). As well the “central management” aspect would be beneficial as it would ensure a common structure and common level of ensured characteristics when compared to IDs created in single agencies (and thus probably using various ID structures of varying uniqueness and persistence).

However, when IDs would be assigned during data upload to the portal and not in their original environment problems can arise from the fact that the data might be uploaded also to other portals (ie a piece of data might get multiple IDs in different portals with no mapping between these) as well as for the synchronisation of data (ie we have to ensure

that when data is rearranged or changed in the original system the ID will remain the same after the data has been re-uploaded to the Archives Portal Europe);

- *Using third party services for assigning IDs:* another possibility is to use third party services which would be independent of any particular infrastructure and thus could be used either at the data providers' location or at Archives Portal Europe. The main benefit of this solution would be that the service provider would coordinate centrally the management and structures of the IDs and therefore ensure that all IDs (regardless of the point of origin) are unique and persistent. However, we can also see that using third party services is not too reasonable in the situation where the data to be identified is not entirely in the control of the portal. Therefore we do not recommend using this scenario.

Looking at the discussion above we recommend implementing a mixed solution, which combines elements from all of the three conceptual scenarios above:

- The Archives Portal Europe defines centrally core use cases and characteristics of supported ID structures. Please note that the current chapter 6 is explicitly intended to support this objective and start necessary discussions;
- More capable data providers are invited to use and create PID matching these requirements on their own;
- If the data providers do not supply an ID on their own or the ID is not meeting the defined characteristics then an "APE PID" is created during data upload, at the same time maintaining the original ID and creating a mapping to the "APE PID";
- The Archives Portal Europe also includes some limited mechanisms and logical algorithms to deal with some crucial management tasks (ie data re-upload, data moving from one archival institution to another, PID forwarding and negotiation with other portals like Europeana and CENDARI etc.);
- If data providers want to, they are also allowed to use third-party PID services but these must also confirm with the Archives Portal Europe requirements, thus from the APE perspective this solution would scale down to the second point in this list (same as data providers creating their own PID).

In short we can see that one of the main roles of the Archives Portal Europe should be the dissemination of best practices about creating and managing persistent identifiers to European archival institutions. In parallel some technical tools need to be available to check the suitability of IDs created by data providers as well as for creating and managing IDs inside the portal if necessary.

6.4. Objects to be identified

6.4.1. Currently identified entities in EAD, EAG and EAC-CPF

As the Archives Portal Europe is relying on the EAD, EAG and EAC-CPF metadata standards it is reasonable to first have a look at the possibilities of these standards in the scope of assigning and using identifiers.

apeEAD

On high level each apeEAD file has an <eadid> element, which identifies a specific finding aid or holdings guide as a whole. The <eadid> and its attributes @mainagencycode for identifying the

institution holding the material (and thus usually having provided the finding aid) and @identifier are mandatory and require content. While the content of <eadid> is up to the content provider and in the majority of cases equals an identifier unique in the originating system, the value for @mainagencycode is generated using a registered ISIL code or an ISIL-like code created by the institution itself respectively taken from the institution's credentials in the Dashboard. The value for @identifier, if not already provided in another way by the content provider, is created during conversion and combines the value of @mainagencycode with the content of <eadid>, thus resulting in an identifier either unique within the system of the Archives Portal Europe (when based on ISIL-like codes for the institution) or globally unique (when based on registered ISIL codes for the institution).

In addition each <c> level includes a <unitid>, which identifies specific description levels (ie fonds, series, items, classes). While the <unitid> is not mandatory many data providers currently use it at least for the highest and lowest description levels. Intermediate, ie mainly structuring levels, might well be enumerated as part of their titles, but mostly do not provide a specified <unitid>.

As such the combination of <eadid> + <unitid> would give us unique identification of all descriptive units but the problem is that not all data providers are able to ensure the persistence of these (especially unitid). The Archives Portal Europe may need to develop a system to check it.

EAG 2012

Each EAG file has a mandatory <recordId> element, which – although mainly meant for identifying the description of an institution – is used to capture the unique identifier of the institution itself, including its repositories as long as these are not handled as independent organisations. Thus, the content of <recordId> is also used in the attribute @repositorycode of the optional <repositorid> element, which originally identifies the whole institution and the repositories. The workflow in the Archives Portal Europe – creation or conversion of EAG documents – ensures, that these two elements have the same content. As an example, the National Archives of Ireland has information in EAG as follows:

<control>

<recordId>IE-NAI</recordId>

...

<archguide>

<identity>

<repositorid countrycode="IE" repositorycode="IE-NAI"/>

It seems wise to use the **recordId** element for identification of the EAG record and it is logical to create PID out of it, as an example: <http://www.archivesportaleurope.net/data/IE-NAI/eag/IE-NAI>

apeEAC-CPF

The approach is similar to apeEAD in the sense that each apeEAC-CPF file has a mandatory <recordId> element, which identifies the whole EAC-CPF file. As each EAC-CPF file describes exactly one authority this could very well be implicitly used to also identify the authorities themselves and not only the description files. According identification could be encoded with the element

<entityId>, which furthermore would be repeatable and allows for the attribute @localType to be used in order to specify the system from where an identifier originates. Thus, it would be possible to eg record the Archives Portal Europe PID next to other entity identifiers such as VIAF or ISNI²⁵.

In addition there is an explicit element available for alternative or historic IDs – <otherRecordId>.

For more information about the identifiers of EAC-CPF you can also refer to chapter 4 in Deliverable D4.4²⁶.

6.4.2. Missing entities

We can see that the available possibilities inside EAD, EAG and EAC-CPF already cater for most of the needs. However, we recommend also assigning IDs to some additional entities, which are not necessarily described in the EAD documents:

- **Digitised records** (as a whole): in many cases descriptions of single records do not exist and therefore there is also no c-level description in the EAD documents. At the same time the single records might be digitised and a link to the digitised record might be available. We can see that there is strong need from users to bookmark and reference to these digitised records and therefore assigning a persistent ID would be recommended. An option for this could be to activate the @id attribute for the <dao> element usually containing the link to digitised records;
- **Single digitised items (pages)**: in addition some digitised records might consist of multiple computer files, most usually single pages in a record. As with the previous item we can see that there is a user need to refer to these single items and we recommend adding persistent identifiers. This would be a case, where the use of METS in combination with EAD could become handy as it allows to more specifically address single digitised items without overloading the EAD documents;
- **Archives Portal Europe ontology or vocabulary values**: In the case that the Archives Portal Europe intends to create own semantic assets then also the values inside these assets need to have persistent IDs, implemented as Persistent Uniform Resource Identifiers (PURI), for LOD purposes;
- **Content created by users**: We might also consider the creation of PID for the tags or any other content created by the users, as well as for the personal collections – link books – which have been created by the users of the portal. However, for now we think that the identification issues next to this kind of data do not necessarily need a persistent and global identifier but are more reasonable to be solved on database level (see also UC13 and UC14 in the next chapter).

6.5. Use cases - using PID for Archives Portal Europe functionality

In this chapter we describe some of the use cases of PID on the portal. While users will often use the PID unnoticed these still play an important role for the functions.

²⁵ International Standard Name Identifier: <http://www.isni.org/> (viewed 2 October 2014).

²⁶ http://www.apex-project.eu/images/docs/APEx_D4.4_SOTA_EAC-CPF_final_version_01.pdf (viewed 2 October 2014).

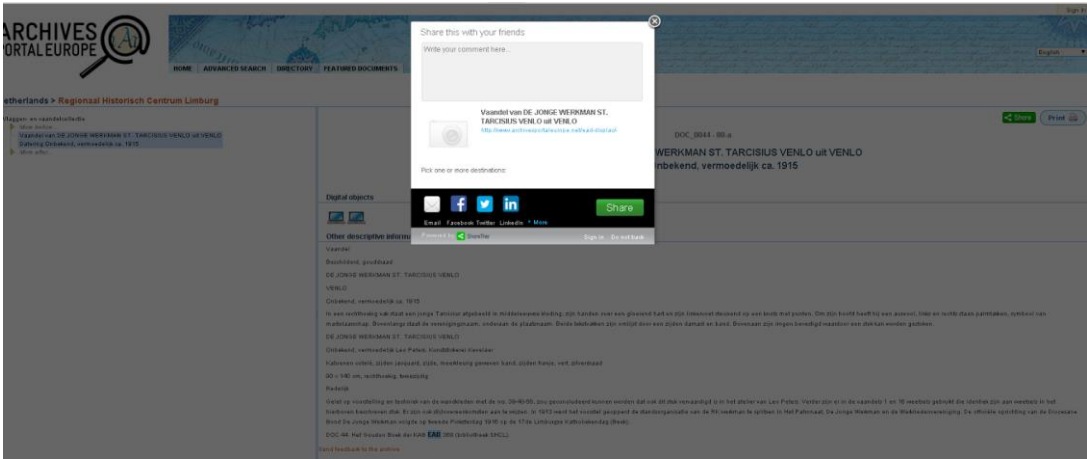
UC9. Linked Open Data (LOD)

Rationale	In LOD triples the subject and object should be possible to be referenced by a URI. Ideally the URI has both a human and machine-readable response available.
Use case	<p>When uploading data to the Archives Portal Europe, a PID should be assigned to the subjects (creator descriptions, descriptive units, also single records and digital objects). Based on this PID a URI should be constructed. For example, data.archivesportaleurope.net/[PID].</p> <p>Ideally, when referring to this URI human users (browsers) should be redirected to an HTML presentation of the subject. For machines an RDF presentation should be delivered.</p> <p>In addition also the objects would “be nice” to be identified by URI, but we might assume the use of external ontologies (eg DBPedia) and their URI/PID schemes for these.</p> <p>Questions to be answered are:</p> <ul style="list-style-type: none"> - Which subjects should be used in the Archives Portal Europe LOD/triples (creators, descriptive units, persons, places); - Are there any ontologies, which the Archives Portal Europe plans to create itself and which components/vocabulary values would need to be assigned PID/URI as well? <p>In the beginning, the scope of creators and descriptive units might be sufficient for LOD identification.</p>

UC10. Citations

Rationale	In academic publishing it is necessary to have all your sources properly cited and, when using resources derived from Archives Portal Europe, PID should be available for this purpose.
Use case	<p>This use case is rather straightforward - users should be able to copy/paste a PID (as a persistent URI or separately) to all descriptive units. In academic world probably other elements (single images, creator descriptions) are less relevant for citations.</p> <p>Some specific requirements for citations could be having a separate button, which copies the persistent URI or PID to the clipboard. The URI should be kept as short as possible, preferably containing the unique PID as the only parameter.</p> <p>Some references for citation (both viewed 2 October 2014):</p> <p>http://www.collectionscanada.gc.ca/005/005-6070-e.html</p> <p>http://researchguides.library.yorku.ca/content.php?pid=324268&sid=2654249</p>

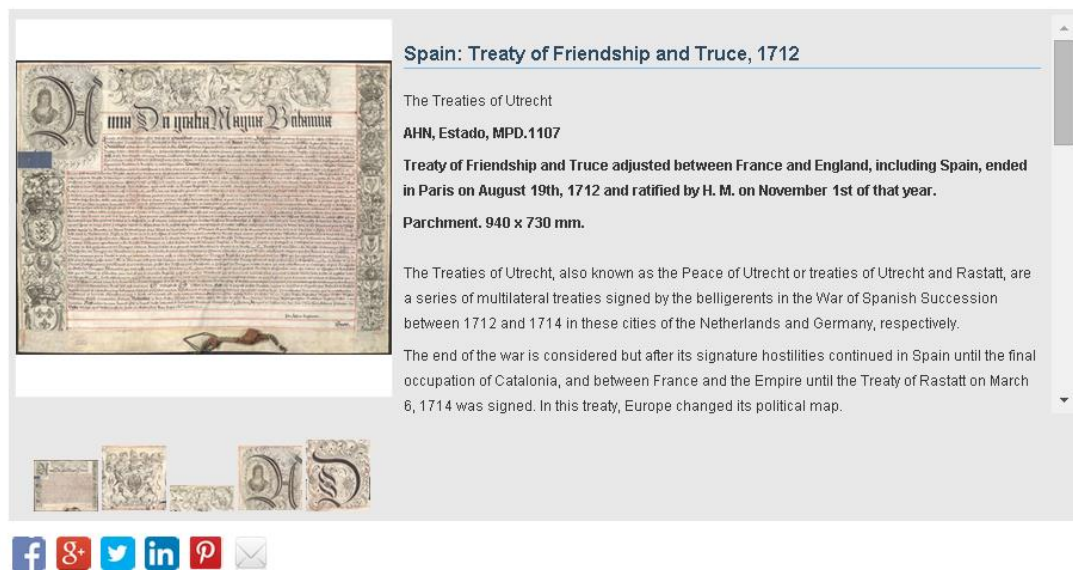
UC11. Share EAG and EAC-CPF data and digital objects with external social network services

<p>Rationale</p>	<p>PID can be used in various media distribution channels such as Facebook and Twitter, where the users can pick up a description, comment on it, and add Like!</p>
<p>Use case</p>	<p>In the 2nd display, the users can mouse-over and select the action (share with Facebook, email, print, PDF, download Zotero format). The users can comment on this at various levels: EAD level and <c> levels in EAD. It should also be made possible for different levels of EAG and EAC-CPF entities.</p> <p>Part of this use case has already been implemented at the time of writing. Users can find a “Share” button at the top right corner of an archival description. When this button is clicked, a pop-up window appears, allowing the user to write a comment and share it by email, Facebook, Twitter, LinkedIn or others. The URI is visible in the window, in light blue. However, at the moment, this option is only available at EAD content.</p> 

UC12. Share virtual exhibition / featured document and digital object identifier with external social network services

<p>Rationale</p>	<p>This use case is much related to Use Case 11, but specialises more on specific single items / digital objects in the Archives Portal Europe.</p> <p>In order to pinpoint a digital object on the portal, it is recommended to have a PID. It allows us to refer to a specific page of the object and/or featured documents page and alike, so that the user does not have to see all EAD information associated with the object.</p> <p>Each object within the featured documents section should have a PID so that users can share them in social media separately from the other objects and exhibition itself.</p>
<p>Use case</p>	<p>A user finds a digital object on the portal and clicks a button next to it to share the information with other people via popular tools including social network services. PID ensures the long-term availability of the shared item on the internet. In case of digital objects in archival descriptions, <dao@id> is the element/attribute for PID use.</p> <p>The sharing of featured documents is partly implemented. The user can share one</p>

featured document in Facebook, Google+, Twitter, LinkedIn, Pinterest and email. However it is not possible to share it image by image in the featured documents, but that should be possible in future.



Spain: Treaty of Friendship and Truce, 1712

The Treaties of Utrecht
AHN, Estado, MPD.1107

Treaty of Friendship and Truce adjusted between France and England, including Spain, ended in Paris on August 19th, 1712 and ratified by H. M. on November 1st of that year.
Parchment. 940 x 730 mm.

The Treaties of Utrecht, also known as the Peace of Utrecht or treaties of Utrecht and Rastatt, are a series of multilateral treaties signed by the belligerents in the War of Spanish Succession between 1712 and 1714 in these cities of the Netherlands and Germany, respectively.

The end of the war is considered but after its signature hostilities continued in Spain until the final occupation of Catalonia, and between France and the Empire until the Treaty of Rastatt on March 6, 1714 was signed. In this treaty, Europe changed its political map.

The main question next to sharing featured documents is how long the “special pages” for featured documents remain available? Having real persistent identification would also come with a commitment to keep the featured documents pages alive forever.

The question next to sharing single images is that these should also be available in the Archives Portal Europe, though not explicitly but instead referenced to the original data provider’s sites (as an example www.ra.ee/jkghkg.jpg and not www.archivesportaleurope.net/jkghkg.jpg). As such it would be really hard for the Archives Portal Europe to ensure the persistency of the links and the easy way out would be to simply use the original URI given by the data provider and thus implicitly assume that all the liability for “broken links” is not with the Archives Portal Europe but with the data provider.

UC13. Crowdsourcing (tagging etc) in the Archives Portal Europe

Rationale	When the Archives Portal Europe starts offering crowdsourcing, commenting and tagging possibilities then such User Generated Content (UGC) might be necessary to be identified separately and also connected to the subjects of crowdsourcing / tagging.
Use case	<p>We assume that the subjects of crowdsourcing (descriptive units, creator descriptions) have already been identified by PID.</p> <p>When users input a specific piece of information this should be automatically linked to the subject in case and (where appropriate) to the specific metadata element in question.</p> <p>In case the crowdsourcing action is based on a formal vocabulary or ontology (geo-tagging/locations) the crowdsourced information should also be identified by a</p>

	<p>persistent URI of the vocabulary value. However, the Archives Portal Europe is in this case only concerned when we develop and use own vocabularies / ontologies and in most cases we should be able to use external vocabularies / ontologies with already established URI structures.</p> <p>The questions in this use case are:</p> <ul style="list-style-type: none"> • Is there any other piece of UGC which needs to be identified persistently (like - single comments, user uploaded images related to a descriptive unit); • Should there be separate PID/URI for a descriptive unit with and without user additions (ie following one you see the “pure” description and the other gives you the enriched information)? <p>However, the assumption for now is that user generated content does not necessarily benefit of having PID as such, with the possible exception of the “my collections” (see UC14).</p>
--	--

UC14. Creating and sharing “my collections”

Rationale	In the Archives Portal Europe logged-in users should be allowed to create personal collections / “my collections” which include references and short descriptions of information in the portal.
Use case	<p>We assume that a link within a collection mostly refers to material, which already has a PID (ie descriptive units, creator descriptions, etc at the Archives Portal Europe).</p> <p>Some links may also refer to external materials (like information available in other portals). Solving the problems around persistent identification of external materials is too difficult and therefore we pragmatically state that the permanent identification of external resources is up to the specific owner of the portal/site).</p> <p>We might also choose to allow users to upload their own content to the collections (including related materials they have digitised themselves). Most probably this content does not need to be identified in a fully persistent manner but an internal (database level) identifier is sufficient. Still, it must be ensured that when the underlying Archives Portal Europe database is migrated to a new technology all relations between the collection and the content uploaded by the user remain valid.</p> <p>Also the collections as a whole need to be possible to be referenced / shared (as an example for inviting “friends” to view or collaborate). Each of the collections “would benefit” of a PID-based URI in order to avoid the problems around platform and configuration changes in the portal.</p> <p>Creating a PID for the whole collection could also be optional, ie the collection PID/URI is only being created/assigned when the collection is shared with another party or is made <i>public</i>.</p> <p>Note: With regard to publishing “My collections” the same question of rights for reusing the descriptive information in new context applies as for the previous use case.</p>

6.6. Use cases - managing PID in the system

In contrast to the chapter 6.5, this chapter will describe the use cases of PID in the system. We, therefore, focus on the configuration side of the story rather than the end-user side, including PID creation and update. For each section, there are two scenarios: for data of the providers and UGC (User Generated Content), because the impact of the PID is different for each case.

UC15. Creating a PID for data delivered by archives

Rationale	It should be possible to reference all published data on the portal using unique global PID. The granularity of the reference should meet the needs of the users. As for apeEAD, it is foreseen that having a PID at file level, <c> level, and <dao> level is considered as highly important. EAG2012 and EAC-CPF will need file and possibly entity level identification.
Use case	PID should be automatically generated when the data is published by dashboard. The portal users would be thus able to refer to the data.
Current URI	<p>EAD file level</p> <p>http://www.archivesportaleurope.net/ead-display/-/ead/pl/aicode/NL-HaNA/type/fa/id/4.AANW</p> <p><c> level</p> <p>http://www.archivesportaleurope.net/ead-display/-/ead/pl/aicode/NL-HaNA/type/fa/id/4.AANW/unitid/4.AANW++2</p> <p><dao> level</p> <p>Not Available in the Archives Portal Europe, though might be as part of the content provider's original system</p> <p>EAG file level</p> <p>http://www.archivesportaleurope.net/directory/-/dir/ai/code/NL-HaNA</p>
Known issues	<c> level URI are generated automatically by Dashboard either based on <unitid> (see discussion on persistence of <unitid> in chapter 6.4.1) or based on a database ID, which is not persistent as they will change when the EAD file is updated.

UC16. Creating PID for User Generated Content

Rationale	PID can be created for all user generated content (UGC) and these can be referred to uniquely and permanently. The user generated contents can be in different forms including a comment, a submitted file, a shared link book, etc.
Use case	When new content is created (ie not content that is re-uploaded and/or edited) in the user's personal workspace and made public, a PID needs to be provided for every new

	object. The PID (meaningful vs. random hash) depends on the underlying PID generation implementation. It has to be ascertained that the newly created PID are unique, to avoid any collisions when using the portal.
Known issues	It is unclear whether PID (not ID) is practical for UGC since it may run into considerably large numbers without securing the quality of UGC making it may be hard to maintain it in the long term.
Proposed solutions	As far as the UGC is related to referenced data on the portal, (ie data with persistent IDs such as EAD, <c> level, and EAC-CPF entity), the dashboard can simply assign a new ID to the UGC. The ID will be automatically created by a certain algorithm. For example, UGC attached to an EAD file (4.AANW) can have the following PID (with SHA-2): http://www.archivesportaleurope.net/data/NL-HaNA/fa/4.AANW/0x730e109bd7a8a32b1cb9d9a09aa2325d2430587ddbc0c38bad911525

UC17. Updating content with PID (for data from archives)

Rationale	When the data to which PID is already assigned, needs to be updated, the portal also has to ascertain the integrity of data and PID.
Use case	<p>When a content provider decides to un-publish data, such data becomes hidden from the public, and therefore the PID is not accessible any more. If the content provider decides to re-publish the data, the same PID will be used and become accessible again.</p> <p>When a content provider decides to delete data completely from the dashboard, the PID becomes inaccessible.</p> <p>On the other hand, when a content provider decides to overwrite the existing published data, the previously assigned PID should be used. When the content provider decides to delete data, and upload the same data afterwards, the portal needs to decide whether to assign a new PID, irrespective of whether the data is exactly the same or not, (re-direction from the old to the new PID may be deployed) or assign the same PID used before.</p> <p>As far as the portal re-uses the local identifier as a part of the PID (ie the current URI for EAD file), it is very likely that the same PID will be assigned (as long as <eadid> is not changed), and therefore, no versioning control is possible. In this case, no system needs to be developed to identify the deleted and newly uploaded files, because <eadid> remains the same and is used as a part of the PID syntax. If a new PID syntax is developed, independent of <eadid>, versioning is possible, but a system needs to be implemented to identify that the two files are originally the same data.</p>
Known issues	<p>Some important decisions have to be made to define the configurations of the PID assignment.</p> <p>As the content providers are free to un-publish data on the dashboard, the PID cannot be guaranteed. In other words, the PID are persistent as far as the data remains</p>

	published. However, on the positive side, the Archives Portal Europe can provide to them the same level of responsibility and control as their local services.
--	--

UC18. Updating content with PID (for User Generated Content)

Rationale	If some material within the portal gets overwritten or updated, versioning should be applied.
Use case	User creates a data object. After some time (s)he decides to update the object with some new data, so a new version of the object is created by the user and a new PID is assigned. Yet, the previous versions of the same document remain intact.
Proposed solutions	As an example, many Wikis and blogs have versioning systems, therefore, when a page is edited, a new link will be provided as a permalink. As UGC commenting and Wiki may be frequently updated by the portal users, so this method seems to be effective for the Archives Portal Europe.

6.7. Summary of PID issues

In this chapter we summarise some of the possibilities and questions next to the generic topic of assigning PID.

Which entities should get PID?

From a highly pragmatic perspective an iterative model should be considered:

- Start with providing PID to the entities which are already been identified (by <eadid>, <unitid> and comparable elements in EAG and EAC-CPF);
- Agree on a URI structure for referencing the entities on the web (something like [www.archivesportaleurope.net/\[mainagencycode\]/\[eadid\]/\[unitid\]/](http://www.archivesportaleurope.net/[mainagencycode]/[eadid]/[unitid]/));
- Provide PID to “my collections” as soon as the functionality becomes available.

With these three actions the most urgent needs should be covered and work can be continued to:

- Provide PID for digitised items / <dao> elements;
- Discuss the PID need for User Generated Content;
- Discuss the need for Archives Portal Europe ontologies / vocabularies and the persistent identification of values in these;

Re-use of PID from provider?

As already mentioned in the very beginning of this document it would be beneficial if we could reuse the persistent identifiers of the original data providers. Unfortunately such a solution cannot be applied in full extent due to lacking technical competence in especially smaller archival institutions – persistent identifiers are often not implemented there and they lack the resources to do so.

The practical solution might be to reuse the ID provided by the data provider if available, if not then a PID might be created by the Archives Portal Europe.

However, as the Archives Portal Europe as such cannot guarantee the persistence of IDs created by individual data providers then:

- We might think about tools to check automatically the conformance of the IDs against the requirements for an Archives Portal Europe PID;
- We might provide tools to check the validity of the ID provided by the institution;
- We might provide concise guidelines to recommend actions and simple steps towards persistency;
- Especially for the data (re)upload scenario simple logical algorithms could be in place to check whether any ID change has happened at the data provider's site.

Constructing PID based on existing identifiers

When reusing already available identifiers we have the following components available, which might be considered:

- Institution identifier (ISIL, includes also the country code);
- <eadid>, <unitid> etc.

The combination of ISIL and <eadid> is used for referencing already now. When information is forwarded to EDM the link to the Archives Portal Europe presentation is like:

<http://www.archivesportaleurope.net/ead-display/-/ead/pl/aicode/NL-HaNA/type/fa/id/4.AANW>
or <http://www.archivesportaleurope.net/ead-display/-/ead/pl/aicode/NL-HaNA/type/fa/id/4.AANW/unitid/4.AANW++2>.

As such the identifier used is "NL-HaNA/4.AANW" for the complete EAD document and "NL-HaNA/4.AANW/4.AANW++2" for one of the components. Something similar ([ISIL] + [eadid] + [unitid]) could ideally also be used as the APE PID.

Available technologies for PID

If the decision is made to use other PID structures then there are quite a few different structures available internationally. Some of these, which we recommend considering, are listed below:

- ***ARK (California Digital Library)***
The Archival Resource Key (ARK), dating from March 2001, is a URL scheme developed at the US National Library of Medicine and maintained by the California Digital Library. ARKs are designed to identify objects of any type – both digital and physical objects. ARK looks like 'http://example.org/ark:/12025/654xz321'. This is clearly an HTTP URI, but it embeds a persistent identifier (ark:/12025/654xz321) inside a URI for an ARK resolver (here http://example.org). While 'ark:/12025/654xz321' is itself an ARK identifier (a name associated with a thing), the URI, which includes a resolver, 'http://example.org/ark:/12025/654xz321', is considered fully equivalent.
- ***PURL***
Persistent Uniform Resource Locators (PURL), proposed in 1995 and developed by OCLC, are

actionable identifiers. A PURL consists of a URL; instead of pointing directly to the location of a digital object, the PURL points to a resolver, which looks up the appropriate URL for that resource and returns it to the client as an HTTP redirect, which then proceeds as normal to retrieve the resource. PURL are compatible with other document identification standards such as the URN. In this sense, PURL are sometimes described as an interim solution prior to the widespread use of URN.

A software package implementing a PURL resolver may be freely downloaded from the OCLC website²⁷. PURL can also be created on the public PURL server.

- **OpenURL**

OpenURL, dating from 2000, contains resource metadata encoded within a URL and is designed to support mediated linking between information resources and library services. The OpenURL contains various metadata elements; the resolver extracts them, locates appropriate services and returns this information. It is sometimes described as a metadata transport protocol.

An OpenURL is formed of a prefix (a valid HTTP URL linking to the user's institutional OpenURL resolver) and a suffix, which is simply a query string encoded according to the URI RFCs, eg [RFC3986](#)-- which deprecates RFC2396, against which OpenURL was initially defined.

- **Digital Object Identifier (DOI)**

The Digital Object Identifier (DOI) was introduced to the public in 1998. The DOI is an indirect identifier for electronic documents based on Handle resolvers. According to the International DOI Foundation (IDF), formed in October 1997 to be responsible for governance of the DOI System, it is a 'mechanism for permanent identification of digital content'.

It is primarily applied to electronic documents rather than physical objects. It has global scope and a single centralised management system. DOI consist of two sections: a numeric identification consisting of a prefix identifying the term as a DOI (10.) and a suffix identifying the document's publisher. The document is then identified with a separate term. The document and publisher are separated by a forward slash, in the format:
doi:10.345/document.identifier12345.

The suffix following the forward slash is either automatically generated by the agency registering the DOI, or is contributed by the registrant. In practice, the suffix is limited to characters that can be encoded within a URL. DOI are not case-sensitive.

In general, no meaning should be inferred to the content of the suffix beyond its use as a unique ID. DOI may be resolved via the Handle system. Although DOI are designed for Unicode-2 (ISO/IEC 10646), the required encoding is UTF-8 due to the fact that the Handle.net resolver uses UTF-8. The DOI is formalised as ANSI/NISO Z39.84-2005, and is currently in the later stages of the ISO certification process.

DOI registration incurs a cost both for membership and for registration and membership of each document, and as such it may in some situations be considered preferable to make use of the Handle.net resolver without the use of DOI.

²⁷ <http://oclc.org/research/activities/purl.html> (viewed 2 October 2014).

- **Handle**

The Handle system was first implemented in 1994 and published as an RFC in November 2003. It is primarily used as a DOI resolver (see example above). In practice, it is a distributed, general-purpose means for identifying and resolving identifiers. Both master and mirror sites are administered by the Corporation for National Research Initiatives (CNRI), and the distributed nature of the service ensures reliable availability. The Handle.net system may also be used separately to the DOI system. The underlying software package may be downloaded and installed for institutional use.

This system was designed by the CNRI, initially with support from the Defense Advanced Research Projects Agency (DARPA).

In addition there is also the possibility to use PID structures, which are based on mathematical algorithms to avoid collisions. The most known of such structures is probably GUID (Globally Unique Identifier - http://en.wikipedia.org/wiki/Globally_unique_identifier, viewed 2 October 2014).

If one of these PID systems is applied we still recommend keeping the original identifier (as provided by the agency) inside the Archives Portal Europe as a secondary identifier.

Maintaining and relating the Archives Portal Europe PID to the PID of the content providers

With creating an alternative ID next to the one, which is possibly already available from the data provider, the question on maintaining the relations between multiple IDs has to be answered.

As a first principle any solution should be simple and straightforward. As a general recommendation the archival standards should be reviewed to allow for documenting any changes in the original IDs. Simply put - when data providers change IDs (<eadid>, <unitid>) it should be recommended (or even mandatory) to also keep the old ID so that relations could be easily checked during data upload to the Archives Portal Europe. As a backup measure also the comparison of other elements (title, dates, upper descriptive levels / position in the hierarchy) might be checked.

7. Linked Open Data

7.1. Overview of issues

As also in D6.1 the terms Open Data (OD) and Linked Open Data (LOD) are in the current document understood as respectively the four or five star open data defined in the “five star open data” model²⁸. This means that:

- **Four star open data** (OD) is data, which is made available on the web as structured data and is using open formats. Each of the data entities is identified by an URI so that other people or applications can reference it. This makes it possible for external actors to reuse the data to create novel mash-up applications.
- **Five star open data** (LOD) has the same characteristics as four star open data but in addition links the data to other resources, especially to global or local ontologies. This makes it possible to automatically build connections to other data sets and therefore add further potential for developing external mash-up applications.

The ultimate goal of the Archives Portal Europe should be to deliver good quality five star data. However, this means that all the content needs to be semantically enriched and links to central semantic assets need to be available. The process of achieving this is not simple and demands a lot of effort, which combines the use of automated tools and manual checks, not least the tagging effort to be concluded.

To have the task of preparing LOD managed in a reasonable way we need to apply a pragmatic and iterative approach. Such an approach has been researched in quite a lot detail in the YEAH! (You, Enhance Access to History!) project²⁹. The project was funded commonly by Vinnova³⁰, NordForsk³¹, the Icelandic Centre for Research (RANNIS)³² and the Estonian Ministry for Economic Affairs and Communications³³. The project ran between January 2012 and April 2014 and the partners included five institutions from Iceland, Sweden and Estonia – the Luleå University of Technology (project coordinator), Estonian Business Archives and the National Archives of Estonia, Sweden and Iceland respectively.

The goal of the project was to investigate the technological components, which could be used to connect and reuse archival holdings with each other and the wider spectrum of e-government data. As the main results all three national archives created LOD demonstrators showing how to provide

²⁸ See also <http://5stardata.info/> (viewed 2 October 2014).

²⁹ <http://www.vinnova.se/sv/Resultat/Projekt/Effekta/2009-04551/YEAH/> (viewed 2 October 2014).

³⁰ <http://www.vinnova.se> (viewed 2 October 2014).

³¹ <http://www.nordforsk.org/en> (viewed 2 October 2014).

³² <http://en.rannis.is/> (viewed 2 October 2014).

³³ <https://www.mkm.ee/en> (viewed 2 October 2014).

novel mash-up type access services using LOD³⁴. Also all the practical experiences have been documented in the project deliverable “Linked Open Data for Memory Institutions: Implementation Handbook”³⁵.

As two of the YEAH! partners are also major contributors to the APE_x project’s LOD effort (National Archives of Sweden and Estonia respectively), the APE_x project has not undertaken in-depth research on that topic but instead we reuse in the next chapter the results of the YEAH! project. Please note that we have got the kind allowance to do so from all other YEAH! partners and have not declared the following as work within the APE_x project.

When looking at the practical implementation of LOD the YEAH! Handbook proposes a six-step workflow as visible on Figure 1.

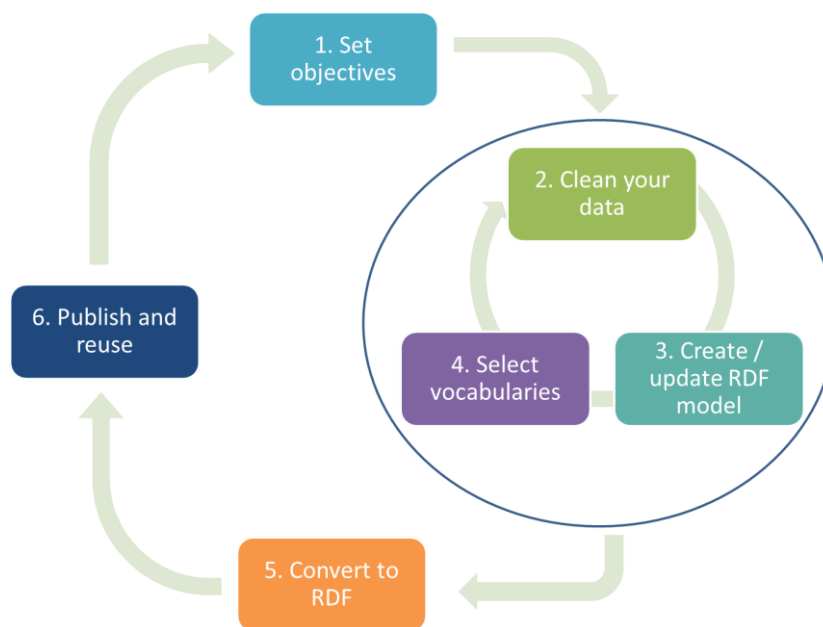


Figure 1: YEAH! Linked Open Data implementation workflow

According to the workflow each LOD project should start with setting clear objectives around the potential use of the LOD created by the (memory) institution. Next you should carry out a set of iterative tasks of cleaning your data, creating an RDF model and selecting appropriate vocabularies. After a few iterations of these tasks you should have a final RDF model available for which you can create appropriate transformation tools and ultimately convert all your data to RDF, which you can then publish for reuse.

The next sub-chapters will look more closely to the application of this model in the context of the Archives Portal Europe.

³⁴ All demonstrators are linked from the project website at <http://www.ltu.se/research/subjects/information-systems/Avslutade-projekt/YEAH-You-Enhance-Access-to-History-1.86175?l=en> (viewed 2 October 2014).

³⁵ https://pure.ltu.se/portal/files/96261604/YEAH_Handbook_ver_1_1_20140506.pdf (viewed 2 October 2014).

7.2. Objectives for archival Linked Open Data

When looking at the needs and reasons why the Archives Portal Europe should support the linking of archival descriptions and the publishing of LOD we can identify the following four high-level objectives:

- Linking archival descriptions available on the Archives Portal Europe to each other and therefore allowing to benefit more out of these than is currently possible;
- Linking archival descriptions better to the content in Europeana;
- Linking archival descriptions to objects in other memory institutions and therefore facilitating the creation of mash-up access solutions. As an example there are multiple thematic and/or regional portals available on topics like World War I, medieval history etc. All of these would benefit from the possibility of reusing the content on Archives Portal Europe more easily;
- Linking archival descriptions to the wider LOD cloud and therefore allowing to mash-up archival descriptions with any external data.

When evaluating these four objectives we can see that especially the first two would not necessarily benefit from an explicit LOD approach. The reason for this statement is that as the archival descriptions are available in common formats on the Archives Portal Europe and Europeana anyway (respectively in the ape formats or EDM) then using an RDF-based LOD scheme would not bring many benefits over simpler linking using the mechanisms inside these internal standards and the according infrastructure.

Simply put – we see that linking data inside Archives Portal Europe or Europeana is easier to be done with other techniques than the RDF/SPARQL based methods in LOD. However, we still see that the semantic enrichment tasks discussed in earlier chapters remain valid when trying to achieve these objectives. As such we can say in a simplified way that we need “Linked Data” in a semantic mapping and description sense but not necessarily “Linked Open Data” as in using open licences and RDF technologies.

Next we can also see that the fourth objective is probably too wide to be practical. Linking archival descriptions to “any information resource available out there” is not reasonable as an initial objective as it would widen the scope of technological and intellectual challenges too much.

As a result our recommendation to the Archives Portal Europe is to concentrate in the LOD activities towards the reuse of archival descriptions in mash-up applications in the cultural sector, while at the same time seeking semantic interoperability between archival institutions themselves and Europeana.

7.3. Preparing your data for publishing and linking it

Based on the discussions presented in the previous chapter we can also state that the backbone for the cleaning and linking of archival data should be the main semantic assets defined already by Europeana. This would allow us to of course fulfil the objective of better interoperability with Europeana but at the same time we can rather safely also assume that the same semantic assets have been or are started to be used by other memory institutions.

As such we can see that the core semantic assets defined for linking by Europeana should also be the core ones for the Archives Portal Europe:

- Places: GeoNames, <http://geonames.org/> (viewed 2 October 2014);
- Concepts: GEMET thesaurus, <http://www.eionet.europa.eu/gemet> (viewed 2 October 2014);
- Persons: DBPedia, <http://dbpedia.org/> (viewed 2 October 2014);
- Time periods: adhoc time period vocabulary provided by Europeana.

In addition the Archives Portal Europe could think about creating a new ontology on archival creators, which would amend and possibly be linked to DBPedia.

Here we would also like to draw your attention to the fact that the actual preparation and linking of data for the provision of LOD is done by the tools and concepts described above in the chapters around the management of semantic assets, user-tagging and NER tools. As such our recommendation would be to align these activities to mainly support the provision of semantically enriched data according to the semantic assets named in this chapter.

7.4. Archival RDF model

In terms of the RDF model, which shall be used to create the technical LOD, the Archives Portal Europe is in a rather good situation.

Namely, all the data, which is uploaded to the Archives Portal Europe, already comes in according to central standards – the apeEAD, EAG 2012 and apeEAC-CPF mark-up standards. As well, a preliminary RDF model for EAD has already been created by the LOCAH project³⁶. This model has been further refined and updated by the UK-based Archives Hub³⁷ and does also include necessary transformation scripts.

At the moment of writing (summer 2014) the Archives Hub is also in the process of becoming a full member of the APE_x project and the Country Manager for the UK. As such the project will be able to benefit from the availability of Archives Hub technical personnel by the end of 2014, which would allow us to review their RDF model. Special attention shall then be turned to the following aspects:

- As the Archives Hub EAD-to-RDF conversion is built on top of the full EAD specification APE_x standards experts need to review the RDF model and the conversion scripts in regard to possible discrepancies between the full EAD and the apeEAD specification;
- Archives Hub has not created RDF profiles for the other sets of data in Archives Portal Europe, namely the apeEAC-CPF and EAG 2012. Though, their practical skills in the area of RDF modelling would certainly speed up the process of generating brand new RDF profiles for these two standards;
- The Archives Hub RDF model has been created before the availability of Europeana Open Data and their EDM model. As discussed above we can see that especially the use of the core Europeana semantic assets shall be an objective for the Archives Portal Europe and therefore also the Archives Hub RDF model needs to be reviewed with that in mind.

³⁶ <http://archiveshub.ac.uk/locah/> (viewed 2 October 2014).

³⁷ <http://archiveshub.ac.uk/> (viewed 2 October 2014).

We hope that the work on providing final RDF profiles for the Archives Portal Europe will be carried out in late 2014/early 2015 and the models with appropriate transformation sheets will be implemented within the Archives Portal Europe Dashboard during 2015.

In addition we would also like to remind you that the semantic enrichment of data on the Archives Portal Europe has not yet started. Therefore we can see that by 2015 most of the data provided by these means would be “four star Open Data” and not LOD. In the best case only very limited amount of test data would be possible to be made available as five star LOD.

7.5. Publishing Linked Open Data at the Archives Portal Europe

The final piece in the LOD puzzle is the actual publishing of LOD at the Archives Portal Europe.

Broadly seen there are two scenarios for that:

- Publish the data as downloadable RDF files;
- Implement a triple store, which would allow querying the triples more easily.

Technically speaking we can see that especially the provision of downloadable RDF files would be rather simple and would not require major developments. Therefore this would be our primary recommendation to the project.

At the same time also the availability of open source triple stores is rather good and implementing one of these at the Archives Portal Europe would not bring much overhead. However, this is still more effort than for the first scenario. Therefore our recommendation here is more careful: the APE_x project should consider the setup of a triple store infrastructure and evaluate the technical needs and possible threats by the end of the project.

Finally we would also like to touch upon two additional items, which need to be thought about:

- Recovery of the LOD: for people to find out about the availability of archival LOD the dataset need to be described in the so-called Open Data Catalogues. Our recommendation for the APE_x project is to set up a new Archival Open Data catalogue inside the Archives Portal Europe infrastructure, which would describe the datasets available, their access possibilities (ie download or directly from the triple store), reuse licensing details and other relevant information. We can also see that the catalogue should not be limited to only data published on the Archives Portal Europe but would be available to also describe datasets which have been published on institutional sites (as an example on the website of a National Archives or a national Open Data portal) or by other means. This catalogue should also be synchronised with other catalogues from relevant memory institutions or portals (ie Europeana);
- Update of LOD: it should also be the aim of the Archives Portal Europe to update the LOD regularly. While automated tools for achieving live-update capability would be possible to be implemented we can see that the difficulty of implementation and the small amount of possible LOD would make this approach unreasonable for now. However, our recommendation to the technical team is to start evaluating possibilities for implementing “RDF update reminders” for data providers who update their EAD data on the Dashboard and which would be followed by manual execution of the RDF conversion. The possibilities for updating the Open Data sets in a more automated manner should be evaluated after the end of the APE_x project in case appropriate funding and/or tools become available.

7.6. LOD Use Cases

UC19. Converting data to RDF

Rationale	In the dashboard, content providers can convert their data to RDF.
Use case	<p>The content provider logs into the dashboard and goes to the content manager section. (S)he can select one or many finding aids and choose the task “Convert to RDF”.</p> <p>The dashboard will automatically check whether the content to be converted, also includes links to central or individual semantic assets. If not, all finding aids will be converted to RDF, which is technically seen as four star open data.</p> <p>If the content includes references or tags directing to central or individual semantic assets, the dashboard will also:</p> <ul style="list-style-type: none"> - Ask whether the central semantic assets should remain as references in the RDF; - Ask whether the individual semantic assets should remain as references in the RDF; - Automatically validate for possible mismatches in the links. <p>A set of RDF files created in this way is called five star open data.</p> <p>A summary report will also be generated, describing how many triples the data includes, how many links to external vocabularies are included, and how many objects have been described.</p>

UC20. Publishing Open Data

Rationale	In the Dashboard, content providers can publish their RDF data.
Use case	<p>The content provider logs into the dashboard and goes to the content manager section, which includes an area for “RDF data”.</p> <p>In this section, the content provider can see a set of finding aids, which have been converted to RDF. (S)he can select them and execute one of the following tasks:</p> <ul style="list-style-type: none"> - Download the RDF files for publishing on their own Open Data sites; - Publish the RDF files on an Archives Portal Europe Open Data site. <p>In case the first option is selected, a simple download procedure will start. In addition, the content provider can choose to describe the published data in the Archival Open Data catalogue, provided by the Archives Portal Europe, as a separate process.</p> <p>In case the second option is selected, as part of the process, license details need to be provided. By default, the content provider will be recommended to use the same licence settings as for Europeana publishing, ie CC0 licence for the data and any of the other possible licences for digitised material. If the content provider has not yet provided any data to Europeana, the CC0 licence will be recommended for everything</p>

by default.

The content provider can modify the licence settings, if needed, and then select to publish the data.

As a result, the RDF files will be:

- Uploaded to the Archives Portal Europe triple store;
- Uploaded to a download site;
- Automatically described on the public Archival Open Data catalogue, provided by the Archives Portal Europe.

The description includes the link to the download, a description on how to address the data on the triple store, and information on licence restrictions.

8. Summary and outlook

The Archives Portal Europe, and in fact also most other aggregators, have been struggling to find the most reasonable way to serve their intended user groups. Already in the early phases of the APE_x project we have described a set of functionalities, which could help users to exploit the information more easily **inside** the Archives Portal Europe. However, we can also see that it is impossible for such a portal to deliver all functionality which some of the user groups interested in European history would need. As an example there are many groups, which are interested in more specific sets of data (like World War I) or functionality (like educational applications, games and tests).

Looking at current technological possibilities we can see that these needs could be addressed by exploiting a (Linked) Open Data approach – publishing the content gathered into the Archives Portal Europe in an open and (technologically) easy to use way which would allow interested parties to select the data they need, mash it up with external resources and provide the necessary access functionality.

This deliverable has looked into the needs and possibilities of achieving this goal and especially concentrated on how the data from various sources and in multiple languages could most efficiently be linked to each other. As a result we have identified that there are some simple steps, which could be investigated with rather minimal effort at least as a proof of concept solution:

- Publishing archival descriptions in RDF format (mainly as four star Open Data);
- Testing the use of NER either on top of EAD or RDF files or on the Archives Portal Europe database;
- Applying simple crowdsourcing and user tagging solutions, which would confirm the information gathered by the NER and therefore create some level of linking between the data.

Achieving these goals on all the content inside the Archives Portal Europe (especially carrying out linking for content in all languages) and delivering production level services is of course too much to do for the remaining six months of the APE_x project. However, we would encourage the project to select some content and at least carry out beta testing to approve the generic concept.

Next to these tasks we also see that there is much to do in the area of raising awareness. Especially the following areas could be addressed as seminars, lectures or written material (like this deliverable) to the content providers:

- Using persistent identification in local catalogues;
- The use of semantic assets in archival descriptions.

Based on the discussions, which hopefully would rise from these actions, the European archival sector could also come to a specific vision on how to manage the global and persistent identification of archival resources. As well, the same would apply to the use of semantic resources and the management of these either inside or outside of the Archives Portal Europe.

When looking more specifically into persistent identification we also need to mention that as this is not only relevant for LOD but also for most other functionalities this issue has already been partly tackled by the APE_x project and by the time of writing a first pragmatic solution has already been applied in the portal. Though, this does not address all the issues which have been highlighted

above in chapter 6 and especially is not always capable of dealing with all situations where data providers themselves change the identifiers. Therefore we strongly encourage continuing discussions and addressing the topic after the end of the project if necessary funding can be secured.

In terms of managing semantic assets the maturity and knowledge of the archival community seems to be least advanced for now. Therefore we repeat here the recommendation to engage the community into discussions on this topic and organise relevant workshops and trainings. As well a best practice guide on managing and using semantic assets could be of benefit along with possible updates to the archival mark-up standards for more advanced support for these assets. In case these actions are carried out we can see potential for 2016 – 2017 to also carry out more practical steps in terms of asset management and semantic mapping functionality in the Archives Portal Europe (as described in chapter 2).

In short – let's start discussing and experimenting already now to achieve a truly useful linked archival information cloud by 2020!

Annex I: Paper evaluation of Named Entity Recognition Tools

Program name	Homepage	Developer	Widely used	Supports multiple languages	Open source	Implementations in memory institutions	Support for a variety of topics / ontologies	Programming language	Remarks
Cogito Discover	http://www.expertsystem.net/products-technology/cogito-discover	Expert System S.p.A.		???	no		part of a suite which supports a wide range of NLP uses	Unknown	
Freeling	http://nlp.lsi.upc.edu/freeling/	Uni Poli de Catalunya		Spanish, Catalan, French, Galician, Italian, English, Russian, Portuguese, Welsh and Asturian. Czech and Slovenian have partial support.	GPL		The Freeling package consists of a library providing language analysis services. See also http://nlp.lsi.upc.edu/freeling/index.php?option=com_content&task=view&id=12&Itemid=41	C++	
GATE	http://gate.ac.uk/	GATE open source community	yes	yes, depending on chosen plugins	LGPL	references include Web Archives	Yes	Java	
Graph Expression	http://code.google.com/p/graph-expression/	huti.ru		en only as it seems, perhaps ru as well	Apache License		a couple:Named Entity Recognition(NER) patterns, optimal match finding (for ambiguous grammars, relation and fact extraction, structure parsing (like document structure, sentence parsing)	Java	
Learning Based Java	http://cogcomp.cs.illinois.edu/page/software_view/11	Cognitive Computation Group, University of		en only	BSD		yes	Java	

Program name	Homepage	Developer	Widely used	Supports multiple languages	Open source	Implementations in memory institutions	Support for a variety of topics / ontologies	Programming language	Remarks
		Illinois							
LingPipe	http://alias-i.com/lingpipe/index.html	Alias-i		yes	royalty free; commercial licence also available		see http://alias-i.com/lingpipe/index.html	Java	
Mallet	http://mallet.cs.umass.edu/	University of Massachusetts		yes, depending on chosen plugins	Common Public License		MALLET is a Java-based package for statistical natural language processing, document classification, clustering, topic modelling, information extraction, and other machine learning applications to text.	Java	
MontyLingua	http://web.media.mit.edu/~hugo/montylingua/	MIT		en only	Free for research		MontyLingua is a free*, commonsense-enriched, end-to-end natural language understander for English	Python, Java	
NLTK	http://www.nltk.org/	International team, see http://www.nltk.org/team.html	would assume so, since there exist various mailing lists and a book	en, es, hi, nl, possibly others depending on the used corpus	Apache License 2.0		NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging,	Java	

Program name	Homepage	Developer	Widely used	Supports multiple languages	Open source	Implementations in memory institutions	Support for a variety of topics / ontologies	Programming language	Remarks
							parsing, and semantic reasoning.		
Nooj	http://www.nooj4nlp.net/pages/nooj.html	University of Franche-Comté		Yes, see http://www.nooj4nlp.net/pages/resources.html	Free for research		Nooj is a linguistic development environment that includes large-coverage dictionaries and grammars, and parses corpora in real time.	.NET-based	
Apache OpenNLP	https://opennlp.apache.org/	Online community		Yes, but don't know which languages are supported	Yes (Apache License, Version 2.0)				
Pattern		Tom De Smedt (tom at organisms.be) Walter Daelemans, University of Antwerp		en, es, de, fr, it, nl	Under BSD licence, Requires : Python 2.5+ on Windows Mac Linux	Especially used in linguistics and psycholinguistics; http://www.clips.ua.ac.be/projects	It doesn't support ontologies. Pattern is a web mining module for the Python programming language. It has tools for data mining (Google, Twitter and Wikipedia API, a web crawler, a HTML DOM parser), natural language processing (part-of-speech taggers, n-gram search, sentiment analysis, WordNet), machine learning (vector space model, clustering, SVM), network analysis and <canvas> visualization. http://www.clips.ua.ac.be/pages/pattern	Pattern is a web mining module for the Python programming language.	
PSI-Toolkit	http://psi-	Adam	doesn't	automatic processing of Polish	GNU	-	The aim of the project is to develop	This description	

Program name	Homepage	Developer	Widely used	Supports multiple languages	Open source	Implementations in memory institutions	Support for a variety of topics / ontologies	Programming language	Remarks
	toolkit.amu.edu.pl/	Mickiewicz University in Poznań	seem	(and - to lesser extent - other languages: English, German, French, Spanish and Russian) with the focus on machine translation	Lesser General Public License		a tool chain (called PSI-Toolkit) for automatic processing of Polish (and - to lesser extent - other languages: English, German, French, Spanish and Russian) with the focus on machine translation No ontologies.	is directed for Ubuntu users, but the installation process for other Linux systems will be similar.	
Rosette	http://www.basistech.com/text-analytics/rosette/	Basis Technology	Yes	Yes.	Commercial	not found	Rosette® is the world's most widely used component library for multilingual text retrieval and analysis. Rosette provides automatic language identification, linguistic analysis, entity extraction, and entity translation from unstructured text, all in a single, unified framework.	C, C++, Java, .NET	
ScalaNLP	http://www.scalanlp.org/	David Hall and Daniel Ramage	doesn't seem	?	Apache licence	not found. Rather mathematical applications are known.	ScalaNLP is the umbrella project for Breeze and Epic. Breeze is a set of libraries for machine learning and numerical computing and natural language processing. Epic is a high-performance statistical parser written in Scala. It uses Expectation Propagation to build complex models without suffering the exponential runtimes one would get in a naive model. Epic is nearly state-of-the-	Scala??? Sbt	

Program name	Homepage	Developer	Widely used	Supports multiple languages	Open source	Implementations in memory institutions	Support for a variety of topics / ontologies	Programming language	Remarks
							part on the standard benchmark dataset in Natural Language Processing.		
Stanford NLP	http://nlp.stanford.edu/software/index.shtml	The Stanford Natural Language Processing Group	Yes	Named Entity Recognition in English, Chinese, and German.	Open source, GNU General Public License (v2 or later)	?	An integrated suite of natural language processing tools for English and (mainland) Chinese in Java, including tokenization, part-of-speech tagging, named entity recognition, parsing, and co-reference.	All the software we distribute here is written in Java. All recent distributions require Oracle Java 6+ or OpenJDK 7+. Much of this software can also easily be used from Python (or Jython), Ruby, Perl, Javascript, and F# or other .NET languages	
Rasp	http://www.sussex.ac.uk/Users/johnca/rasp/index.html	University of Cambridge, University of Sussex	? Probably not	No	Open source, under the GNU Lesser General Public License.	not known	RASP is a domain-independent, robust parsing system for English.	C++	

Program name	Homepage	Developer	Widely used	Supports multiple languages	Open source	Implementations in memory institutions	Support for a variety of topics / ontologies	Programming language	Remarks
Natural	https://github.com/NaturalNode/natural	Copyright (c) 2011, 2012 Chris Umbel, Rob Ellis, Russell Mull	Not	At the moment, most of the algorithms are English-specific, but in the long-term, some diversity will be in order. Thanks to Polyakov Vladimir, Russian stemming has been added!, Thanks to David Przybilla, Spanish stemming has been added!.	Yes, WordNet licence	not known	Natural is a general natural language facility for node.js. Tokenizing, stemming, classification, phonetics, tf-idf, WordNet, string similarity, and some inflections are currently supported. It's still in the early stages, so we're very interested in bug reports, contributions and the like.	JavaScript, NodeJS	
Text Engineering Software Laboratory (Tesla)	http://tesla.spinfo.uni-koeln.de/index.html	developed at the Department of Computational Linguistics at the University of Cologne, Germany	Not known, it rather seems to be a software for linguistic laboratories	? This needs a deeper search into the documentation. It seems that the framework is language independent, so it's a more generic framework.	THE ACCOMPANYING ECLIPSE PUBLIC LICENSE	not known	Tesla is a virtual research environment for text engineering - a framework you can use to create experiments in corpus linguistics, and to develop new algorithms for natural language processing. Tesla is a client-server application, which can be used by individual researchers as well as by workgroups. Pos tagging, NER,	Written in Java, Currently adding support for Python and Scala.	Tesla is mostly a laboratory for computational linguistics, not a framework for repetitive workflow execution.
Thinktelligence Delegator	http://www.thinktelligence.com/	Thinktelligence Corporation	probably not	probably not	Commercial	not known	The delegator is an API that lets you add commands in natural languages to your programs user interface.	SDK implemented in Java	doesn't seem to be appropriate for our purpose
Treex	http://ufal.mff.cuni.cz/treex/	Institute of Formal and Applied	probably not	Czech, English	Open source, Free	probably not	It is primarily aimed at Machine Translation, making use of the ideas and technology created during the	Perl under Linux	

Program name	Homepage	Developer	Widely used	Supports multiple languages	Open source	Implementations in memory institutions	Support for a variety of topics / ontologies	Programming language	Remarks
		Linguistics (ÚFAL) at the Computer Science School, Faculty of Mathematics and Physics, Charles University in Prague, Czech Republic			licence		Prague Dependency Treebank project. At the same time, it is also hoped to significantly facilitate and accelerate development of software solutions of many other NLP tasks, especially due to re-usability of the numerous integrated processing modules (called blocks), which are equipped with uniform object-oriented interfaces.		
UIMA	http://uima.apache.org/index.html	Apache	yes (http://uima.apache.org/index.html , Events and Conferences)	Yes	Apache licensed open source	not known, it would need more research	Unstructured Information Management applications are software systems that analyze large volumes of unstructured information in order to discover knowledge that is relevant to an end user. An example UIM application might ingest plain text and identify entities, such as persons, places, organizations; or relations, such as works-for or located-at.	Java and C++	
VisualText	http://www.textanalysis.com/	Text Analysis International, Inc.	Moderately, http://www.textanalysis.com/Cust	?	FREE for personal, internal, academic, develop	not known	VisualText is the premier integrated development environment for building information extraction systems, natural language processing systems, and text analyzers.	Analyzers developed with VisualText can run stand-alone or embedded in larger	

Program name	Homepage	Developer	Widely used	Supports multiple languages	Open source	Implementations in memory institutions	Support for a variety of topics / ontologies	Programming language	Remarks
			omers/customer s.html		ment, and non-commercial use.		Information Extraction Shallow Extraction Intelligent Web Crawlers Indexing Categorization Text Mining Summarization Automatic Coding Natural Language QueryText from Speech Text to XML, SQL	applications on most computers that run C++ (e.g., Linux).	
WebLab-project	http://weblab-project.org/index.php?title=WebLab	OW2 Consortium	Moerately yes, http://weblab-project.org/index.php?title=Publications	Yes >30	Yes, (under LGPL 2.1)	not known	The WebLab is aimed at providing intelligence systems and any other applications that need to process multimedia data (text, image, audio and video). Data acquisition (Web, data bases, folders, TV, radio...) Normalization of content (text, image, videos...) Language identification (> 30 languages) Speech-to-text transcription Annotation and sources assessment Named entities extraction in texts Object and concept detection in image and videos Semantics analysis Relation extraction Thematic categorization and clustering Automatic summarization Indexing Full Text	Java	

Program name	Homepage	Developer	Widely used	Supports multiple languages	Open source	Implementations in memory institutions	Support for a variety of topics / ontologies	Programming language	Remarks
							Search (keywords, annotation, boolean, etc.) Semantics search Information mapping		
Unitex/GramLab	http://www-igm.univ-mlv.fr/~unitex/	Laboratoire d'Informatique Gaspard-Monge	Yes http://www-igm.univ-mlv.fr/~unitex/index.php?page=10 Quite long bibliography: http://www-igm.univ-mlv.fr/~unitex/index.php?page=10	Yes	open source, GNU LESSER GENERAL PUBLIC LICENSE	not known	Unitex is a corpus processing system, based on automata-oriented technology. The concept of this software was born at LADL (Laboratoire d'Automatique Documentaire et Linguistique), under the direction of its director, Maurice Gross. With this tool, you can handle electronic resources such as electronic dictionaries and grammars and apply them. You can work at the levels of morphology, the lexicon and syntax. The main functions are: building, checking and applying electronic dictionaries pattern matching with regular expressions and recursive transition networks applying lexicon-grammar tables handling ambiguity via the text automaton aligning texts building an automaton from a certified corpus	C++ (Core components) & Java (Visual IDE)	It rather seems to be a linguistic software without NER options, but it contains quite a lot of dictionaries.
The Dragon Toolkit	http://dragon.ischool.drexel.edu/	Drexel University	?	?	GPL	not known	The Dragon Toolkit is a Java-based development package for academic	Java	

Program name	Homepage	Developer	Widely used	Supports multiple languages	Open source	Implementations in memory institutions	Support for a variety of topics / ontologies	Programming language	Remarks
							<p>use in information retrieval (IR) and text mining (TM, including text classification, text clustering, text summarization, and topic modelling). It is tailored for researchers who work on large-scale IR and TM and prefer Java programming. Moreover, different from Lucene and Lemur, it provides built-in supports for semantic-based IR and TM. The dragon toolkit seamlessly integrates a set of NLP tools, which enable the toolkit to index text collections with various representation schemes including words, phrases, ontology-based concepts and relationships. However, to minimize the learning time, we intentionally keep the package small and simple. The toolkit does not have some features including distributed IR and cross-language IR which is a part of Lemur toolkit.</p> <p>http://dragon.ischool.drexel.edu/features.asp</p>		
Palladian	http://palladian.ws/	Dresden University of Technology	?, e.i. http://webknox.c	Yes, language classifier can be trained. But the	The complete source	probably not	Palladian is a Java-based toolkit, which provides functionality to perform Internet Information	Java	Algorithms can be adapted from this tool. It has a clear

Program name	Homepage	Developer	Widely used	Supports multiple languages	Open source	Implementations in memory institutions	Support for a variety of topics / ontologies	Programming language	Remarks
			om		code is licensed under the Apache License 2.0 Source code available		Retrieval tasks such as crawling, classification, and extraction of various types of information. It provides a collection of algorithms for text processing focused on classification, extraction, and retrieval. The aim of Palladian is to reuse algorithms that are freely available and build upon them to drive research by providing unified interfaces. This way, new algorithms can be quickly compared to the state-of-the-art allowing other users to create more advanced programs in the future. Palladian is not a full natural language processing suite nor does it contain a full set of algorithms in the fields of classification, extraction, and information retrieval. http://palladian.ws/downloads/palladian-0.10-documentation.pdf		documentation.
Factorie	http://factorie.cs.umass.edu/index.html	UMass Center for Intelligent Information Retrieval	Probably not	?	Factorie has been released under the Apache	probably not	FACTORIE has been successfully applied to various tasks in natural language processing and information integration, including named entity recognition entity	Java, (Apache Maven and Scala required)	Good documentation

Program name	Homepage	Developer	Widely used	Supports multiple languages	Open source	Implementations in memory institutions	Support for a variety of topics / ontologies	Programming language	Remarks
					License 2.0, and is free to use for commercial or academic purposes. Sources available.		resolution relation extraction parsing schema matching ontology alignment latent-variable generative models, including latent Dirichlet allocation.		
Silpa Indic Language Processing Toolkit	URL wasn't found	Silpa open source community developers			AGPL			Python?	
Text Extraction, Annotation and Retrieval Toolkit	https://github.com/louis Mullie/treat	personal developers (Louis Mullie?)	probably not	?	This software is released under the GPL License and includes software released under	probably not	Treat is a toolkit for natural language processing and computational linguistics in Ruby. The Treat project aims to build a language- and algorithm- agnostic NLP framework for Ruby with support for tasks such as document retrieval, text chunking, segmentation and tokenization, natural language parsing, part-of-speech tagging, keyword extraction and named entity recognition.	Ruby	It doesn't seem to be very "live". The latest update is about 4 months and most of the updates are older than a year or so.

Program name	Homepage	Developer	Widely used	Supports multiple languages	Open source	Implementations in memory institutions	Support for a variety of topics / ontologies	Programming language	Remarks
					the GPL, Ruby, Apache 2.0 and MIT licences.				
Zihuita NLP API	doesn't exist	not found	probably not	?	Free for research		I can't found information.	C	The website disappeared
YAGO / AIDA	http://www.mpi-inf.mpg.de/yago-naga/yago/	Max Planck Institute for Technology	Not sure	Yes	CC-BY-SA 3.0	not known	yes, Wikipedia, WordNet, GeoNames	Java	YAGO is the semantic knowledge base, which is mainly used in the application AIDA, though there are others. Therefore we use the common term YAGO here.

Annex II. Closer evaluation of Named Entity Recognition tools

Evaluation results for GATE

Basic data

Name of tool: GATE - General Architecture for Text Engineering

Homepage: <http://gate.ac.uk/>

Developer: GATE project

Licence: CC BY-NC-SA

List of current users (if available, memory institutions preferred):

- ARCOMEM (UK): memory institutions in the e-Social age
- The National Archives (UK): Bringing semantic annotation to the UK government's web archive.
- Perseus (USA): The Perseus digital library, one of the largest and most advanced such projects in the world, uses GATE for corpus annotation and language processing.

Programming languages / frameworks used:

Software platform: Java

Input format: Plain Text, HTML, SGML, XML, RTF, Email, PDF (some documents), Microsoft Office (some formats), OpenOffice (some formats), UIMA CAS, CoNLL/IOB

Output format: same as input

Configuration (Java descriptor XML)

Interface (command line, Eclipse GUI, Java programming API, REST web API, HTML GUI)

Installation sources (SVN, Maven, Eclipse repository, code.google.com)

Main characteristics / components

“GATE supports documents in a variety of formats including XML, RTF, email, HTML, SGML and plain text.” (<http://gate.ac.uk/sale/tao/>) (viewed 17 April 2014)

“[...] the main purpose of GATE is annotating documents. Whilst applications can be used to annotate the documents entirely automatically, annotation can also be done manually, eg by the user, or semi-automatically, by running an application over the corpus and then correcting/adding new annotations manually.” (<http://gate.ac.uk/sale/tao/>) (viewed 17 April 2014)

Download from web server or subversion repository from the provider only. It is delivered with a platform-specific installer, which installs the software easily on the computer. No administration rights needed.

“GATE will run anywhere that supports Java 7 or later, including Solaris, Linux, Mac OS X and Windows platforms. We don't run tests on other platforms, but have had reports of successful installs elsewhere.” (<http://gate.ac.uk/sale/tao/>) (viewed 11 July 2014)

“GATE components are one of three types:

- LanguageResources (LRs) represent entities such as lexicons, corpora or ontologies;
- ProcessingResources (PRs) represent entities that are primarily algorithmic, such as parsers, generators or ngram modellers;
- VisualResources (VRs) represent visualisation and editing components that participate in GUIs.

[...] Collectively, the set of resources integrated with GATE is known as CREOLE: a Collection of REusable Objects for Language Engineering.” (<http://gate.ac.uk/sale/tao/>) (viewed 11 July 2014)

“GATE is distributed with an IE system called ANNIE, A Nearly-New IE system (developed by Hamish Cunningham, Valentin Tablan, Diana Maynard, Kalina Bontcheva, Marin Dimitrov and others). ANNIE relies on finite state algorithms and the JAPE language [...]. ANNIE components form a pipeline which appears in figure 6.1.” (<http://gate.ac.uk/sale/tao/>) (viewed 11 July 2014)

Ontology support

Supported European languages, which are or (are going to be) in the portals apeEADs: French, English, German, Italian, Spanish, (Bulgarian, Romanian, Russian).

“GATE provides an API for modelling and manipulating ontologies and comes with two plugins that provide implementations for the API and several tools for editing ontologies and using ontologies for document annotation.” (<http://gate.ac.uk/sale/tao/>) (viewed 11 July 2014)

Hands-on evaluation

Installation process

Easy download from GATE website and with platform-specific installer makes installation on PC smooth. Special configurations are possible, but not necessary.

Sending data to the tool

Loading files is easy like open a file in any other software. User can open single documents or create documents collections. No API needed. Configurations are possible, but not necessary.

Files can be loaded only one-by-one and are added to document collects one-by-one. Batch processing might be possible with DataStore function, which wasn't tested..

Test data used: English EAD: <http://www.archivesportaleurope.net/ead-display/-/ead/pl/aicode/GB-00000002402/type/fa/id/Add+MS+88938>

Receiving and reusing results

Output can be specified. Human readable: statistic summary and text with (without XML tags) with highlighted annotations and possibility to correct this resp. make new ones.

Output can be saved as XML, which exports nodes only. Furthermore, it can be saved in the original format, like EAD, as well.

GATE provides datastore function, which creates either Lucene Based Searchable DataStore or SerialDataStore: file-based storage using Java serialisation. This datastore function was not tested so far.

Quality of results

ANNIE default setting:

File	all annotations	locations/wrong (xpath in output file) number of entities	organisations/wrong number of entities	persons/wrong (xpath in output file) number of entities
English	232,250	533/327 100/21	169/44 107/26	564/50 273/33
French	not working at all			

Performance of the tool

- English file: 608 KB loaded in 0,516 sec
 - ANNIE with default settings working: approx. 43 sec (--> 648 KB with annotations)
- French file: 32,206 KB loaded in 21,828 sec; 15,232 KB loaded in 14,875 sec; 5,121 KB 2,953 sec
 - French NE with default settings working: not working
 - shorted French file with 2,828 KB was finally processed; bug message for GATE appeared
 - “gate.util.LuckyException: Congratulations, you found the ONLY bug in GATE!”

Final verdict

Running GATE on Windows cannot be recommended. Assuming, that GATE was made for less structured documents like plain text or at most TEI-XML, it seems that EAD is too much for the software. Loading files and processing results in bad respond time for the software and every other running software on the PC. Parallel work is not possible.

Should be tested in other environment or with modified configurations.

A big advantage is the XML structure for single applications, which makes it easy to personalise them or to add own word lists. NER results can surely be improved with modifications in the settings of the used plug-ins.

Evaluation results for Stanford NLP

Basic data

Name of tool: Stanford NLP toolkit. (Stanford CoreNLP, Stanford Parser, Stanford POS Tagger, Stanford Named Entity Recognizer, Stanford Word Segmenter, Stanford Classifier, Tregex, Tsurgeon, and Sengrex; Phrasal; Stanford English Tokenizer; Stanford TokensRegex; Stanford Temporal Tagger (SUITime))

Homepage: <http://www-nlp.stanford.edu/index.shtml>

Developer: The Stanford Natural Language Processing Group

Licence: Open source, **licensed under the GNU General Public License** (v2 or later). Note that this is the *full* GPL, which allows many free uses, but *does not allow* its incorporation into any type of distributed proprietary software, even in part or in translation. **Commercial licensing** is also available.

Software platform: MS Windows, Linux (tested on CentOS)

Input format: TXT, CSV, XML, list of files (TXT, CSV, XML)

Output format: XML, Tagged texts files (TXT, CSV, XML)

Configuration: Using Stanford CoreNLP, it is usual to create a configuration file (a Java Properties file). Minimally, this file should contain the "annotators" property, which contains a comma-separated list of Annotators to use. For example, the setting below enables: tokenization, sentence splitting (required by most Annotators), POS tagging, lemmatization, NER, syntactic parsing, and co-reference resolution. (Java descriptor XML)

Interfaces: command line, GUI, Java programming API, and a lot of other third party wrappers to several NLP modules/components (See below).

Installation sources: SVN

Main characteristics / components

Please describe here the main components and functionalities of the tool, hope that you'll be able to fill this by copy-paste from the homepage:

- which actions are possible to be carried out (simple NER, any other NLP tasks, ontology management and creation etc);
 - **Stanford CoreNLP** (An integrated suite of natural language processing tools for English and (mainland) Chinese in Java, including tokenization, part-of-speech tagging, named entity recognition, parsing, and co-reference.)
 - **Stanford NER** (A Conditional Random Field sequence model, together with well-engineered features for Named Entity Recognition in English, Chinese, and German.)
 - **Stanford Parser** (Implementations of probabilistic natural language parsers in Java: highly optimized PCFG and dependency parsers, a lexicalized PCFG parser, and a deep learning re-ranker.)

- **Stanford POS Tagger** (A maximum-entropy (CMM) part-of-speech (POS) tagger for English, Arabic, Chinese, French, and German, in Java.)
 - **Stanford Word Segmenter** (A CRF-based word segmenter in Java. Supports Arabic and Chinese.)
 - **Stanford Classifier** (A machine learning classifier, with good feature templates for text categorization. Provides Naive Bayes and a conditional loglinear classifier (a.k.a., a maximum entropy or multiclass logistic regression model).)
 - **Stanford Tregex, Tsurgeon, and Sengrex** (A utility for matching patterns in linguistic trees (following the tgrep/tgrep2 tradition) and a tree-transformation utility built on top of this matching language. Also, a similar utility for matching patterns in dependency graphs.)
 - **Stanford Phrasal** (A state-of-the-art phrase-based machine translation system.)
 - **Stanford English Tokenizer** (A fast tokenizer for English text (producing Penn Treebank tokenization, roughly))
 - **Stanford TokensRegex** (A tool for matching regular expressions over tokens.)
 - **Stanford Temporal Tagger** (A rule-based temporal tagger for English text.)
- is it available as download and local installation or can only be used by a web API or are both ways possible:
Stanford NLP is available as download and it can be used by web API as well.
 - list any 3rd party components, which the tool uses (relevant mainly for “framework” style tools - like GATE - which actually implement tools provided by others).

Extensions: Packages by others using Stanford CoreNLP

Annotators/models

- A stopword removal annotator by John Conwell.
- GATE Twitter part-of-speech tagger, including [a pos.model loadable by CoreNLP](#).

Java

- [cleartk-stanford-corenlp](#) is a **UIMA** wrapper for Stanford CoreNLP built by Steven Bethard in the context of the [ClearTK](#) toolkit.
- [dkpro-core-gpl](#) is a collection of NLP components, principally Stanford CoreNLP, wrapped as **UIMA** components, based on work at the Ubiquitous Knowledge Processing Lab (UKP) at the Technische Universität Darmstadt. It is part of the [DKPro](#) project. See also the [DKPro Core wiki](#) and [a tutorial on the Stanford CoreNLP components](#).
- [A Vert.x module for accessing Stanford CoreNLP](#) by Jonny Wray.
- [Wrapper for each of Stanford's Chinese tools](#) by Mingli Yuan.

- [RESTful API for integrating between Stanford CoreNLP and Apache Stanbol](#) by Rupert Westenthaler and Cristian Petroaca.

Thrift

- [Apache Thrift server for Stanford CoreNLP](#) by Diane Napolitano. (Written in Java, but usable from many languages.)

C#/F#/.NET

- [Stanford CoreNLP for .NET](#) by Sergey Tihon. (See also: [NuGet page](#).)

Python

- [A Python wrapper for Stanford CoreNLP](#) by Chris Kedzie (see also: [PyPI page](#))
- [An up-to-date fork of Smith \(below\) by Hiroyoshi Komatsu and Johannes Castner](#) (see also: [PyPI page](#)).
- [Updated fork of Smith \(below\) by Robert Elwell](#).
- [Original Python wrapper including JSON-RPC server by Dustin Smith](#).

Ruby

- [Ruby bindings](#) by Louis Mullie (see also: [Ruby Gems page](#)).
- The larger [TREAT](#) NLP toolkit by Louis Mullie also makes available Stanford CoreNLP.

Perl

- [Perl wrapper](#) by Kalle Räisänen.

Scala

- [Scala API for CoreNLP](#) by Mihai Surdeanu, one of the original developers of the CoreNLP package.

Clojure

- [Clojure wrapper for CoreNLP](#) by Cory Giles. Incomplete. Currently only a parser wrapper.

- [Clojure wrapper for CoreNLP](#) by Nils Grünwald. Incomplete. Currently only wraps tagger and TokensRegex.

JavaScript (node.js)

- [stanford-simple-nlp](#) is a node.js CoreNLP wrapper by xissy
- [stanford-corenlp-node](#) is a webservice interface to CoreNLP in node.js by Mike Hewett

Extensions: Packages by others using Stanford NER

For many (computer) languages, there are more up-to-date interfaces to Stanford NER available by using it inside [Stanford CoreNLP](#), and you are better off using those...

- **UIMA:** Florian Laws made a Stanford NER [UIMA](#) annotator using a modified version of Stanford NER, which is available on his [homepage](#). *[Old version.]*
- **Perl:** Kieren Diment has written [Text-NLP-Stanford-EntityExtract](#), a Perl module that provides an interface to Stanford NER running as a server.
- **Ruby:** tiendung has written [a Ruby Binding](#) for the Stanford POS tagger and Named Entity Recognizer.
- **Python:** Dat Hoang wrote [pyner](#), a Python interface to Stanford NER. *[Old version.]* [NLTK \(2.0+\)](#) contains an interface to Stanford NER written by Nitin Madnani: [documentation](#) (note: set the character encoding or you get ASCII!), [code](#), [on Github](#).
- **F#/C#/.NET:** Sergey Tihon has [ported Stanford NER to F# \(and other .NET languages, such as C#\)](#), using IKVM. See [his blog post](#) or [its listing on NuGet](#).

Ontology support

Included with Stanford NER are a 4 class model trained for CoNLL (Computational Natural Language Learning), a 7 class model trained for MUC (Message Understanding Conference), and a 3 class model trained on both data sets for the intersection of those class sets:

- 3 class: Location, Person, Organization
- 4 class: Location, Person, Organization, Misc
- 7 class: Time, Location, Organization, Person, Money, Percent, Date

These models each use distributional similarity features, which provide some performance gain at the cost of increasing their size and runtime. Also available are the same models missing those features.

Also available, as part of a package of caseless models for several of our tools, are caseless versions of these same three models. You can either unpack the jar file or add it to the classpath; if you add the jar file to the classpath, you can then load the models from the path `edu/stanford/nlp/models/...` It can run `jar -t` to get the list of files in the jar file.

A few German models are available. Chinese models are also provided that are built from the Ontonotes Chinese named entity data. There are two models, one using distributional similarity clusters and one without. These are designed to be run on *word-segmented Chinese*.

New classifiers can be quite easily constructed.

The main steps of constructing a new classifier are the following:

- Tokenizing a given text.
(ie `java -cp stanford-ner.jar edu.stanford.nlp.process.PTBTOKENIZER test2.txt > test2.tok`)
- Defining entities within the tokenized text.
(ie `perl -ne "chomp; print qq{$_\t0\n}" test2.tok > test2.tsv`)
- Creating a new classifier. In the creation process the tsv file will be used as train file applied to a given text file. Both the train file and the text file with many other options can be given in a property file.
(`java -cp stanford-ner.jar edu.stanford.nlp.ie.crf.CRFClassifier -prop mol.prop`)

e.i: mol.prop:

#location of the training file

trainFile = test2.tsv

#location where you would like to save (serialize to) your

#classifier; adding .gz at the end automatically gzips the file,

#making it faster and smaller

serializeTo = test2.ser.gz

#structure of your training file; this tells the classifier

#that the word is in column 0 and the correct answer is in

#column 1

map = word=0,answer=1

#these are the features we'd like to train with

#some are discussed below, the rest can be

#understood by looking at NERFeatureFactory

useClassFeature=true

useWord=true

useNGrams=true

#no ngrams will be included that do not contain either the

#beginning or end of the word

noMidNGrams=true

useDisjunctive=true

maxNGramLeng=6

```
usePrev=true
useNext=true
useSequences=true
usePrevSequences=true
maxLeft=1
#the next 4 deal with word shape features
useTypeSeqs=true
useTypeSeqs2=true
useTypeySequences=true
wordShape=chris2useLC
```

Languages: English and algorithms for processing Arabic, Chinese, and German text.

Other than English, the developer currently provides trained CoreNLP models for Chinese. To run CoreNLP on Chinese text, you first have to download the models, which can be found in the [release history](#).

Include this .jar in your classpath, and use the StanfordCoreNLP-chinese.properties file it contains to process Chinese. For example, if you put the .jar in your distribution directory, you could run (adjusting the .jar date file extensions to your current release):

```
java -cp stanford-corenlp-YYYY-MM-DD.jar:stanford-chinese-corenlp-YYYY-MM-DD-models.jar -Xmx3g
edu.stanford.nlp.pipeline.StanfordCoreNLP -props StanfordCoreNLP-chinese.properties -file your-chinese-file.txt
```

We tried to apply the tokenization and classifier generalization to German and Hungarian texts, but the special German and Hungarian accents of the words were cut. Probably only because we couldn't set up the right input and output character sets, so this still needs further investigation.

Hands-on evaluation

Installation process

The installation steps are clearly specified on the website. We downloaded the installation packages of the Stanford CoreNLP and Stanford NER modules. Both modules require Java 1.6 or later. We installed Java 1.7 v. onto a Linux (CentOS) server and onto 3 PC with Windows 7. OS. The Stanford installation packages had to be simply copied onto the computer (after extracting the compressed file).

The Stanford CoreNLP could be started on the Linux server only in command line mode.

Unfortunately on two of four PCs the installed Java environment was not satisfactory for the Stanford CoreNLP, so the Stanford CoreNLP could be launched only on two PCs.

We installed Stanford NER onto a server with Linux (CentOS) and onto a PC with MS Windows 7 operation system.

We could launch NER by command line mode and by GUI on the server and on the PC as well.

Stanford NER system can be set up to allow single-jar deployment

Stanford NER can be run as a server/service/servlet too.

Sending data to the tool

Stanford CoreNLP

By command line mode to Linux server installed program:

We sent simple text (txt, English, 1,9 Kbytes) and XML (apeEAD, English, 622 Kbytes and Hungarian 161 Kbytes) files to analyze.

```
java -cp stanford-corenlp-3.3.1.jar:stanford-corenlp-3.3.1-models.jar:xom.jar:joda-time.jar:jollyday.jar:ejml-0.23.jar -Xmx3g edu.stanford.nlp.pipeline.StanfordCoreNLP -ner -file inputfile_name
```

It is possible sending to analyse a list of files too.

Stanford NER

By command line mode to Linux server installed program:

We sent simple text (txt, English, 42 Kbytes) and XML (apeEAD, English, 622 Kbytes and Hungarian 161 Kbytes) files to analyze.

Through GUI to on the PC installed program:

We sent simple text (txt, English, 42 Kbytes) and XML (apeEAD, English, 622 Kbytes and Hungarian 161 Kbytes) files to analyze.

Receiving and reusing results

Stanford CoreNLP

By command line mode to Linux server installed program:

We sent simple text (txt, English, 1,9 Kbytes) and XML (apeEAD, English, 622 Kbytes and Hungarian 161 Kbytes) files to analyze.

The result was an XML file (the original name and the XML extension). In case of 1,9 Kbytes simple English text file the size of the result was 280 Kbytes. In the case of the 161 Kbytes Hungarian apeEAD export file the size of the result was XXXX. In the last case the result couldn't be await.

Stanford NER

Through GUI to on the PC installed program:

The result was the original files (txt, XML) with new tags.

For example:

- <PERSON>William Shakespeare</PERSON> was the son of <PERSON>John Shakespeare</PERSON>, an alderman and a successful glover originally from <LOCATION>Snitterfield</LOCATION>, and <PERSON>Mary Arden</PERSON>, the daughter of an affluent landowning farmer.[9] He was born in <LOCATION>Stratford-upon-Avon</LOCATION> and baptised there on 26 April 1564. His actual date of birth remains

unknown, but is traditionally observed on 23 April, Saint <PERSON>George</PERSON>'s Day.[10] This date, which can be traced back to an 18th-century scholar's mistake, has proved appealing to biographers, since <PERSON>Shakespeare</PERSON> died 23 April 1616.[11] He was the third child of eight and the eldest surviving son.[12]

- <scopecontent><p>
 A magyar kancellária elnökségén intézett, kálifelnősen bizalmasnak tekintett kormányszati és kifizetési ügyekre vonatkozó iratok. Az elnökségen az 1827-1836 és 1845-1848. évek között iktattak titkosan is, az állagban tehát ezekből az évekből vannak iratok. Nagyjében megegyezik ez a kottát írt Revitzky Ádám és <PERSON>Apponyi</PERSON> György kancellársága évkönyveivel, ami azt tanúsítja, hogy a kávéntek az elnöki állagon kávéllal a második, titkos iktatást. Az állag tárgya egyébként hasonlít az A 45 (<PERSON>Acta</PERSON> praesidialia) állagához. A 100. évfolyamra iktatászmok rendjében fekszenek az iratok. Iktatás és mutatásnyv az egysz állaghoz rendelkezésre áll. 1828 június és július hónapban Revitzky kancellár a szliácsi főről helyen is foglalkozott a fontosabb kancelláriai <PERSON>Álgyekkel</PERSON>. Ezeket ott kálifelnő iktattatta a "Protocollum itinerale" elnevezésű iktatásnyvbe. Az iktatászmok alatt az R betű áll, ami lehet a Revitzky név kezdőbetűje, de lehet a <MISC>Reise</MISC> vagy a <ORGANIZATION>Reservata</ORGANIZATION> szárvényvidése.
 Szliácsi és intézett iratok az 1848. évi iratai után, a 19. csomót végén helyezkednek el. Az iktatásnyv az 1848. évi iktatásnyv végén található.
 </p></scopecontent>
- <scopecontent encodinganalog="summary">
 <head>Scope and Content</head>
 <p>Various items, including: Certificates of <LOCATION>Birth</LOCATION>, Baptism and Marriage; papers concerning <PERSON>Ballard</PERSON>'s schooling in <LOCATION>Shanghai</LOCATION> and <LOCATION>Cambridge</LOCATION>; two passports; some family letters and cards; correspondence with 10 <ORGANIZATION>Downing Street</ORGANIZATION> concerning the award (declined) of CBE; certificate from De <ORGANIZATION>Montfort University</ORGANIZATION> conferring Doctor of Letters; and a number of photographs. </p>
 </scopecontent>

Quality of results

In case of English texts the results were very good. Persons, locations, organizations were recognized all in the texts (at least we didn't find unrecognized objects).

In case of Hungarian text the result was very poor. There were many (high portion) not recognized and also misinterpreted entities.

The typical problems were the following:

- When place name occurred in an organization name or in a person name, NER usually recognized only the place name part of an organization name. The same happened when a place name occurred in a personal name.
- When a Hungarian accented character occurred in the word.

- When a personal name occurred in an organization name. Usually only the personal name were recognized.
- The entity recognition in Hungarian texts at this given level is not good enough to be applied in the APE_x Portal.

Performance of the tool

Stanford CoreNLP:

By command line mode to Linux server installed program:

The performance is very low. Even the result file of the 1,93 kbytes input file appeared only half hour after the process was started.

Stanford NER:

By command line mode to Linux server installed program:

The result was sent to the screen. The recognition was fast, but writing the result into an output file hasn't been tested yet.

Through GUI to on the PC installed program:

The performance and the creation of output files with all input files (txt, English, 1,9 Kbytes, English/622 Kbytes/ and Hungarian/160 Kbytes/) were quite fast.

Application of the NER GUI to XML and TXT test files

Test files	Classifiers								
	english-3		english-4		english-7		english-3-2		
	quality	speed	quality	speed	quality	speed	quality	speed	
XML	english-apeEAD_Add_MS_88930_EADPortal.xml	Organization, Location, Person (2-3 errors)	few seconds	2-4 errors/page, can't distinguish between tags and ordinary text	few seconds	various, 0-8 errors/page, can't distinguish between tags and ordinary text, specific errors are	20 seconds	2-3 errors/page	few seconds
	italian-APEX_IT-ASRA-F750340065.xml		the file is too large to be tested		the file is too large to be tested		the file is too large to be tested		the file is too large to be tested
	french-FRAD06_03E en EAD_v20120321.xml		the file is too large to be tested		the file is too large to be tested		the file is too large to be tested		the file is too large to be tested
TXT	short English text (1031 chars)	No error	1 sec	1-2 errors (organization)	1 sec	No error	1 sec	No error	1 sec
	long English text (4317 chars)	1-2 errors (organization/location)	1-2 sec.	1-2 errors (organization/location)	1-2 sec.	1-2 errors (organization/location)	1-2 sec.	1-2 errors (organization/location)	1-2 sec.
	short French text (2051 chars)		1 sec		1 sec		1 sec		1 sec
	long French text (3660 chars)	person detection was incorrect	1-2 sec.	person detection was incorrect	1-2 sec.	correct location detection	1-2 sec.	person detection was incorrect	1-2 sec.
	short German text (1396 chars)	half of the detections of persons are incorrect	1 sec	half of the detections are incorrect	1 sec	seems to be good	1 sec	half of the detections are incorrect	1 sec
	long German text (3909 chars)	half of the detections are incorrect	1-2 sec.	almost all entity detectiona are incorrect	1-2 sec.	almost all entity detectiona are incorrect	1-2 sec.	almost all entity detectiona are incorrect	1-2 sec.
	short Italian text (1068 chars)	only one misinterpreted entity was detected	1 sec	5 detections, but 4 of them seem contain error	1 sec	4 detections, 3 dates are correct, 1 organisation seems to be correct but other entities were not detected.	1 sec	3 detections, all are wrong	1 sec
	long Italian text (7329 chars)	half of the detections of persons are incorrect	1-2 sec.	half of the detections are incorrect	1-2 sec.	seems to be good	1-2 sec.	2-3 detection of persons and organizations are incorrect	1-2 sec.

Evaluation results for Tool UIMA

Basic data

Name of tool: Apache UIMA framework

Homepage: <http://uima.apache.org/>

Developer: Apache

Licence: Apache ver. 2 open source

List of current users: software developers & integrators (most of them not domain-specific)
<https://cwiki.apache.org/confluence/display/UIMA/Powered+by+Apache+UIMA>

Software platforms: framework - Java, C++; add-ons: Java, C/C++, Perl, Python, TCL

Input formats: TXT, dictionary: CSV, XML

Output formats: XML (XMI)

Configuration: Java descriptor XML

Interfaces: command line, Eclipse GUI, Java programming API, REST web API

Installation sources: SVN, Maven central repository, code.google.com, uima.apache.org, Eclipse Update Site

Evaluation environment: 64 bit Windows 7 SP1

Main characteristics / components

UIMA (Unstructured Information Management applications) are software systems that analyze large volumes of unstructured information in order to discover knowledge that is relevant to an end user. An example UIM application might ingest plain text and identify entities, such as persons, places, organizations; or relations, such as works-for or located-at.

UIMA enables applications to be decomposed into components, for example "language identification" => "language specific segmentation" => "sentence boundary detection" => "entity detection (person/place names etc.)". Each component implements interfaces defined by the framework and provides self-describing metadata via XML descriptor files. The framework manages these components and the data flow between them.

UIMA building blocs accordingly to OASIS standard

(See also <http://docs.oasis-open.org/uima/v1.0/cd01/uima-spec-cd-01.html>)

UIMA *Analysis Engine* (AE) is a program that analyzes artefacts (eg documents) and infers information from them. Analysis Engines are constructed from building blocks called *Annotators*. An *Analysis Engine* (AE) may contain a single annotator or it may be a composition of others and therefore contain multiple annotators.

An *annotator* is a component that contains analysis logic. Annotators analyze an artefact (for example, a text document) and create additional data (metadata) about that artefact. It is a goal of UIMA that annotators need not be concerned with anything other than their analysis logic – for example the details of their deployment or their interaction with other annotators.

Annotators produce their analysis results in the form of typed *Feature Structures*, which are simply data structures that have a type and a set of (attribute, value) pairs. An annotation is a particular type of *Feature Structure* that is attached to a region of the artefact being analyzed (a span of text in a document, for example).

For example, an annotator may produce an Annotation over the span of text President Bush, where the *type* of the *Annotation* is Person and the *attribute* fullName has the *value* George W. Bush, and its *position* in the *artefact* is character position 12 through character position 26.

It is also possible for annotators to record information associated with the entire document rather than a particular span (these are considered *Feature Structures* but not *Annotations*).

All feature structures, including annotations, are represented in the UIMA *Common Analysis Structure* (CAS). The CAS is the central data structure through which all UIMA components communicate.

The CAS is the shared structure between the components. It is made of a

- *Sofa*, subject of analysis, which can be either a String (eg to stand for HTML content), a Binary data (eg to stand for an audio signal), or merely a URL. and the
- *index of metadata*, which result from the analysis engines.

Type System is a schema or data class model for the CAS. It defines the types of objects and their properties (or features) that may be instantiated in a CAS. A specific CAS conforms to a particular type system. UIMA components declare their input and output with respect to a type system. Type Systems include the definitions of types, their properties, range types (these can restrict the value of properties to other types) and single-inheritance hierarchy of types.

UIMA add-ons

UIMA is a framework mediating sub-components as add-ons.

NLP NER add-ons:

- wrappers for Apache OpenNLP, GATE, LingPipe and other external annotators
- *Concept Mapper Annotator* <https://uima.apache.org/d/uima-addons-current/ConceptMapper/ConceptMapperAnnotatorUserGuide.html>
- Dictionary Annotator - <https://uima.apache.org/sandbox.html#dict.annotator>
- JULIE Lab Named Entity Tagger - http://www.julielab.de/Resources/Software/NLP_Tools.html
- OpenCalais Annotator <http://www.opencalais.com/>
- RegularExpression annotator
- Tika annotator
- WEKA Machine learning annotator

Other NLP add-ons

- document text parser - extracts the plain text from a document in various file types (Word, PDF, plain text, HTML, XML) using some open source libraries like PDF box or NekoHTML.
- whitespace tokenizer - simple whitespace tokenizer that extracts tokens from a plain text document for whitespace separated languages.

- language detection - annotator that detects the language of a document using for examples simple language specific word lists.

UIMA integration and performance components:

- UIMA Ruta (Rule-based Text Annotation)
- Scale-out Frameworks UIMA-AS, UIMA-DUCC
- Integration of annotation results into search index - Lucene CAS indexer (Lucas), Apache SOLR CAS
- casToXML - provides a UIMA CAS consumer that writes the analysed documents in a configurable XML representation to the file system.

PEAR packaging is used for simpler installation of add-ons.

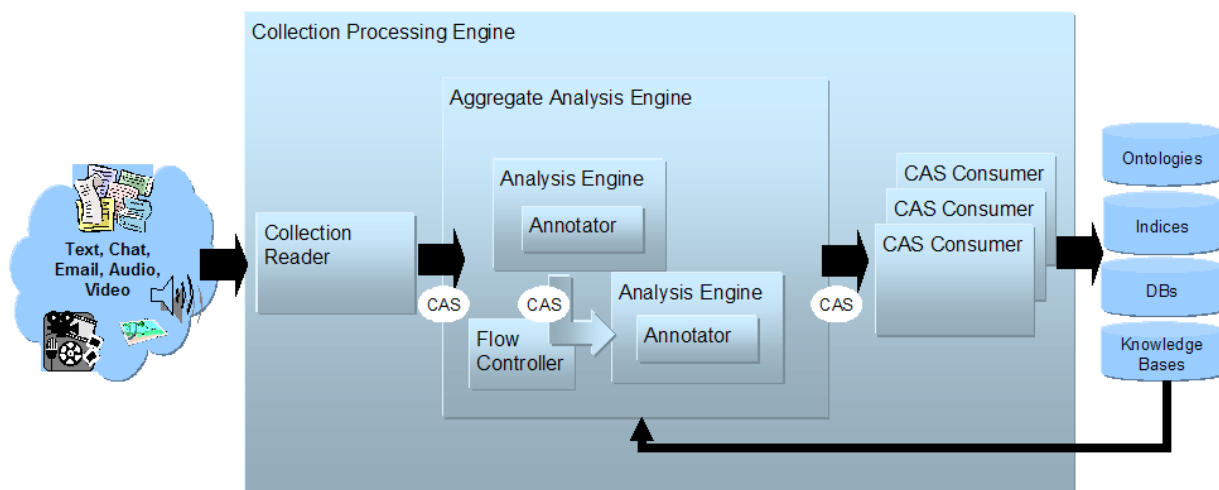


Figure 2: UIMA general architecture (image source: uima.apache.org)

Figure 1 describes which UIMA components must be integrated for any particular UIMA application: CollectionReader for input, AnalysisEngine to select annotators and dictionaries to be used for input processing and CAS Consumer for output presentation.

General UIMA apply process:

1. Define TypeSystem
2. Define AnalysisEngine descriptor
3. Implement Annotator(s)
4. Execute the UIMA pipeline

Example annotator for deploy – Dictionary Annotator

The UIMA dictionary annotator aims at recognizing in an annotation feature path text entries defined in CSV or XML dictionaries. The dictionary can be declared as a resource or locally via a parameter. It is stored in memory as a prefix tree to parse it quickly. The columns of the CSV dictionary can be associated on the fly to a feature structure either independently to feature of a specified annotation or as a whole for a string array annotation feature. Users community

describes Dictionary Annotator as simplest to deploy and Concept Mapper Annotator as the most complex UIMA annotator (deployment requires programming).

Ontology support

UIMA supports essentially English but is not limited with any particular ontologies or languages. UIMA contains tools to create new dictionaries/ontologies and thus includes machine-learning features for support of different languages. UIMA NER annotator enables to recognize and suggest new ontology records according to predefined text pattern.

Hands-on evaluation

Installation process

Lot of installation possibilities and tutorials about installation exist, so for new users it is difficult to find the simplest combination to use or what minimum set of UIMA source files is required:

1. Installation of UIMA Java source code: <http://uima.apache.org/one-time-setup.html> or detailed description <https://cwiki.apache.org/confluence/display/UIMA>. These tutorials suggest to install: Java RE+SDK, UIMA Java-source files (>1.4 GB), SVN client, Maven, Eclipse and integration plug-ins. User will be guided to programming, debugging, and building of the code and Java applications from scratch.
2. Installation of Eclipse UIMA-tools – see <http://uima.apache.org/documentation.html>. Usage of Eclipse for annotator application <https://uima.apache.org/doc-uima-annotator.html> These tutorials suggest to start install with Java RE+SDK, Eclipse tools as UIMA Eclipse-update files (~40 MB), Eclipse EMF tools. The tutorials go shortly over to (Java) developers' guides.
3. Installation of UIMA Java binary files (~40 MB) contain 20 command-line shell files and many jar-files per each 13 add-ons. This solution is fastest and simplest to apply, tutorial can be found at <http://uima.apache.org/doc-uima-examples.html>

Installation summary: as UIMA development history starts from 2005, the amount of project artefacts has grown to level where it is difficult to keep clear starting point for newcomers. The main target group seems to be Java developers with programming and Eclipse usage skills. Any demo-site does not exist to make an inspiration.

Good introduction to UIMA is given in book *Introduction to Linguistic Annotation and Text Analytics* By Graham Wilcock

<http://books.google.ee/books?id=TDQJb1UgVywC&lpg=PA117&ots=bB1a5ZQSTx&dq=uima%20dictionary%20annotator%20pear&pg=PA107#v=onepage&q=uima%20dictionary%20annotator%20pear&f=false>

Sending data to the tool

Two Java desktop GUI components exist to estimate UIMA functionality:

- *DocumentAnalyser* (documentAnalyzer.bat) enables to select one input and one output folder, one annotator descriptor file and see annotated text. Input data can be in text or xmi file format, output is shown on screen and stored as xmi file.
- *CollectionProcessingEngine* (cpeGui.bat) enables to select a CollectionReader descriptor file describing analysis input, one or many AnalysisEngines (ie annotator add-ons) descriptors for performed text processing and CAS Consumer descriptor for analysis output.

The testing goal is to get working UIMA Dictionary Annotator - the simplest UIMA annotator to start with and which enables NER functions.

The two .bat components are working with example (tutorial) annotators and settings, but do not run included add-on annotators. I tried to run DictionaryAnnotator with different configurations, classpath options etc. but the result was similar: “annotator class file not found”. My Java knowledge is not sufficient to debug source files to find and fix errors.

One solution exists – the use of PEAR packaging and PEAR installer for integration of UIMA core and annotation components. Annotator files must be packaged into PEAR format (command line option was used for that:

```
runPearPackager.bat -compID DictionaryAnnotator -mainCompDesc C:\_temp\UIMA\uima_bin\apache-
uima\addons\annotator\DictionaryAnnotator\desc\DictionaryAnnotator.xml -mainCompDir
C:\_temp\UIMA\uima_bin\apache-uima\addons\annotator\DictionaryAnnotator\lib\ -targetDir
C:\_temp\UIMA\uima_bin\apache-uima\addons\
```

Then the created pear file must be installed into UIMA file folder, runPearInstaller.bat script provides GUI for the installation. “Installation ended successfully”, unfortunately the result

NER analyser set-up process:

1. select annotator add-on (there DictionaryAnnotator)
2. prepare dictionary (ontology) to be used as reference system (UIMA includes CSV->XML converter for simple dictionary creation)
3. run DocumentAnalyser or CollectorProcessingEngine bat file
4. select input, output folders and analyser descriptor to be used
5. run analysis

Receiving and reusing results

Results are in human (Java or HTML GUI) and machine readable (XMI-XML) formats. Additional input-output APIs can be developed.

Dictionary entities relations to the analysed text are identified in output XML by unique key value (in case of Dictionary Annotator usage).

Quality of results

Not tested

Performance of the tool

Not tested

Final verdict

Apache UIMA enables rich set of features and is highly scalable (additional annotators can be programmed and added into framework), dictionaries can be added or created by learning. As UIMA is widely used quite long time, the framework should be proven its value, but in APE project time-frame it seems not realistic to apply because of very long learning-curve of UIMA integration and set-up.

Evaluation results for Tool NERD

Basic data

Evaluation environment: 64 bit Windows 7 SP1, Mozilla Firefox 29.0.1, Opera 12.17

Name of tool: NERD API (Named Entity Recognition and Disambiguation)

Homepage: <http://nerd.eurecom.fr/>

Developer: EURECOM Graduate School and Research Center (France)

Licence: limited freeware (not for commercial, benchmarking or competitive use)

List of current users (if available, memory institutions preferred): -

Software platforms: N/A

Input formats: plain text, web page URL

Output formats: XML, JSON

Configuration: included in query syntax

Interfaces: graphical user interfaces on web pages of many related annotators exist but not for NERD framework itself, REST Web API with POST and GET requests, Java, Python, Nodejs and Ruby clients.

Installation sources: github

Main characteristics / components

NERD is a web application plugged on top of various NLP tools. Its architecture follows the REST principles and provides a web HTML access for humans and an API for computers to exchange content in JSON or XML.

Supported annotators:

- AlchemyAPI
- dataTXT (Dandelion)
- DBpedia Spotlight
- Lupedia
- OpenCalais
- Saplo
- SemiTags
- TextRazor
- THD (Targeted Hypernym Discovery)
- Wikimeta
- Yahoo! Content Analysis
- Zemanta

NERD provides one common syntax for different annotator components.

Annotators enable features:

- entity recognition and tagging,
- recognition of related texts,
- contextual analysis,
- text categorization,
- language detection,
- sentiment analysis.

The API interface is developed following the REST principles and aims to enable programmatic access to the NERD framework. GET, POST and PUT methods manage the requests coming from clients to retrieve the list of NERs, classification types and URIs for a specific tool or for the combination of them. They take as inputs the URI of the document to process and a user key for authentication. The output sent back to the client can be serialized in JSON or XML depending on the content type requested.

The REST engine runs on Jersey and Grizzly technologies. Their extensible framework allows to develop several components, so NERD is composed of 7 modules, namely: authentication, scraping, extraction, ontology mapping, store, statistics and web. The authentication enables to log in with an OpenID provider and subsequently attaches all analysis and evaluations performed by a user with his profile. The scraping module takes as input the URI of an article and extracts its main textual content. Extraction is the module designed to invoke the external service APIs and collect the results. Each service provides its own taxonomy of named entity types it can recognize. The ontology mapping is the module in charge to map the classification type retrieved to the NERD ontology. The store module saves all evaluations according to the schema model we defined in the NERD database. The statistic module enables to extract data patterns from the user interactions stored in the database and to compute statistical scores such as Fleiss Kappa and precision/recall analysis. Finally, the web module manages the client requests, the web cache and generates the HTML pages. *(cited from 1)*

Ontology support

NERD ontology (RDF-namespace, <http://nerd.eurecom.fr/ontology>) simplifies NERD usage in Linked Open Data context. The ontology contains also links to widespread RDF-namespaces.

The ontology core classes are:

- Thing
- Amount
- Animal
- Event
- Function
- Location

- Organization
- Person
- Product
- Time

Supported dictionaries. NERD annotators support no user-specific dictionaries but only 'built-in dictionaries' - most annotators are integrated with DBpedia/Wikipedia dictionary, some additional dictionaries are supported by annotators:

- AlchemyAPI:
 - DBpedia
 - CERN website
 - Freebase
 - US Census
 - GeoNames
 - UMBEL
 - OpenCyc
 - YAGO
 - MusicBrainz
 - CIA Factbook
 - CrunchBase
- TextRazor API:
 - DBpedia
 - Freebase
- Zemanta API:
 - DBpedia
 - traileraddict
 - IMDB
 - Amazon
 - DcComics

Supported languages depend on selected annotator, mostly: English, Spanish, French and German. TextRazor supports 10 languages.

Table. Summary about main characteristics of NERD annotators is given in next table (source - 3):

	AlchemyAPI	DBpedia Spotlight	Extractiv	OpenCalais	Zemanta
Language Support	English, French, German, Italian, Portuguese, Russian, Spanish, Swedish	English, Spanish, Portuguese	English	English, French, Spanish	English
Entity type number	272	272	6	39	81
LOD Dataset number	7	1	1	9	1

Table. Equivalent classes among the most frequent categories (source - 3):

AlchemyAPI	DBpedia Spotlight	Extractiv	OpenCalais	Zemanta
Continent	Continent	CONTINENT	Continent	-
Country	Country	COUNTRY	Country	-
City	City	CITY	City	-
Mountain	-	MOUNTAIN	-	-
Lake	Lake	LAKE	-	-
Company	Company	-	Company	company
Person	Person	PERSON	Person	person
Athlete	Athlete	-	-	-
Politician	Politician	-	-	-
BasketballPlayer	Basketball Player	-	-	-
Movie	Film	MOVIE	Movie	film
Automobile	Automobile	-	-	-

Hands-on evaluation

Installation process

Web API does not need any installation when web REST API is used via browser, otherwise Java, Python, Nodejs or Ruby client is used. To start using NERD you first need to get *NERD API Key* by registering your account. Query addressing is very straight-forward.

Sending data to the tool and receiving of responses

Annotating process in general is as follows: upload document to be annotated (text or uri) > run annotation > request results (entities). Next chapters describe the process in more details. (Technical description for programmable API clients can be found at <http://nerd.eurecom.fr/api/application.wadl>)

Document request parameters (UTF8)

POST <http://nerd.eurecom.fr/api/document>

key	The NERD APIkey.
text	The text file, which will be processed to extract entities. Although the field is optional, it is required if {timedtext,uri} are not declared.
timedtext	The SRT file, which will be processed to extract entities. Although the field is optional, it is required if {text,uri} are not declared.
uri	The URI of the article. Although the field is optional, it is required if {timedtext,text} are not declared.

Response

idDocument	The document identifier.
------------	--------------------------

Example

POST Request

```
curl -i -X POST http://nerd.eurecom.fr/api/document -d
"uri=http://www.bbc.co.uk/news/world-us-canada-19644448&key=YOUR_API_KEY"
```

Response

```
{ "idDocument":164 }
```

Annotation request parameters (UTF8)

POST http://nerd.eurecom.fr/api/annotation

key	The NERD APIkey.
idDocument	The document identifier.
extractor	The name an extractor. The accepted values are: {combined, alchemyapi, datatxt, dbspotlight, lupedia, opencalais, saplo, semitags, textrazor, thd, wikimeta, yahoo, zemanta}.
[ontology]	The accepted values are: core, extended. The default value is core.
[timeout]	The maximum interval in seconds to perform the annotation.

Response

idAnnotation	The id of the document.
--------------	-------------------------

Example

POST request

```
curl -i -X POST http://nerd.eurecom.fr/api/annotation -d
"key=YOUR_API_KEY&idDocument=164&extractor=alchemyapi&ontology=core&timeout=10"
```

Response

```
{ "idAnnotation":427 }
```

Entity request parameters

GET http://nerd.eurecom.fr/api/entity

key	The NERD APIkey.
idAnnotation	The annotation identifier.
[granularity]	Accepted values: oen / oed. The oen (One Entity per Name) reads all the entities found in the document. The oed (One Entity per Document) removes duplicates (a duplicate happens when two or more entities have the same NE,type and URI) and reads only one occurrence.

Response

Array(entity)	An array of entity object. The extractor field assumes the following values: alchemyapi, datatxt, dbspotlight, opencalais, lupedia, saplo, semitags, wikimeta, yahoo, zemanta (names of the services supported) or combined. For further details, see the example below.
---------------	--

Example

GET request

```
curl -i -X GET -H "Accept: application/json"
"http://nerd.eurecom.fr/api/entity?key=YOUR_API_KEY&idAnnotation=427"
```

Response

```
[
  {
    idEntity: 120,
    label: "BBC",
    startChar: 138,
    endChar: 141,
    extractorType: "Company",
    nerdType: "http://nerd.eurecom.fr/ontology#Organization",
    uri: "http://dbpedia.org/resource/BBC",
    confidence: 0.0582796,
    relevance: 0.5,
    extractor: "dbspotlight"
  },
  ...
]
```

Quality of results

NERD framework does not have a graphical user interface for testing (although many NERD-related annotators have) and running of REST web-API requests was not possible due to insufficient programming skills. Thus references to articles about quality estimations are given instead.

1. Giuseppe Rizzo, Raphaël Troncy, ***NERD: a framework for unifying named entity recognition and disambiguation extraction tools***. Published in Proceeding: EACL '12 Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics; Pages 73-76, Association for Computational Linguistics Stroudsburg, PA, USA ©2012 (online - <http://dl.acm.org/citation.cfm?id=2380936>)

Evaluation results

Table 3. Number of axioms aligned for all the tools involved in the comparison according to

the NERD ontology based on the same sample data set. (Note that there are listed also Evri and Extractiv that are not any more supported by NERD).

	AlchemyAPI	DBpedia Spotlight	Evri	Extractiv	OpenCalais	Zemanta
Person	6,246	14	2,698	5,648	5,615	1,069
Organization	2,479	-	900	81	2,538	180
Country	1,727	2	1,382	2,676	1,707	720
City	2,133	-	845	2,046	1,863	-
Time	-	-	-	123	1	-
Number	-	-	-	3,940	-	-

- Giuseppe Rizzo, Marieke van Erp, Raphaël Troncy, **Benchmarking the Extraction and Disambiguation of Named Entities on the Semantic Web**. In 9th International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, May 26-31, 2014 (online - http://www.eurecom.fr/~troncy/Publications/Rizzo_Erp_Troncy-lrec14.pdf)

Evaluation results:

Figure 1: Precision, Recall and F-measure results for individual NERD extractors, Stanford and NERD-ML on CoNLL-2003 Reuters data set for different classes and overall. The black line denotes the upper limit the combined extractors can obtain.

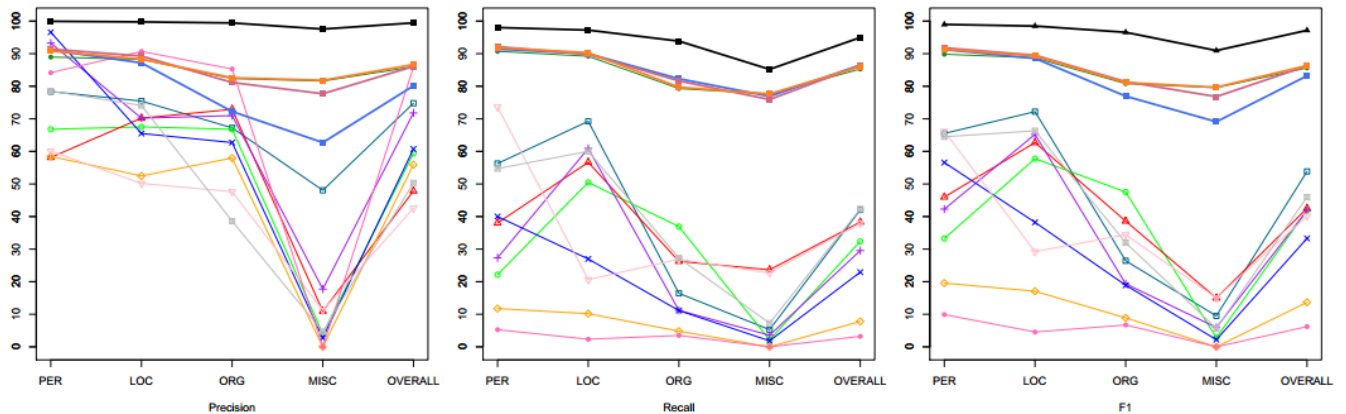
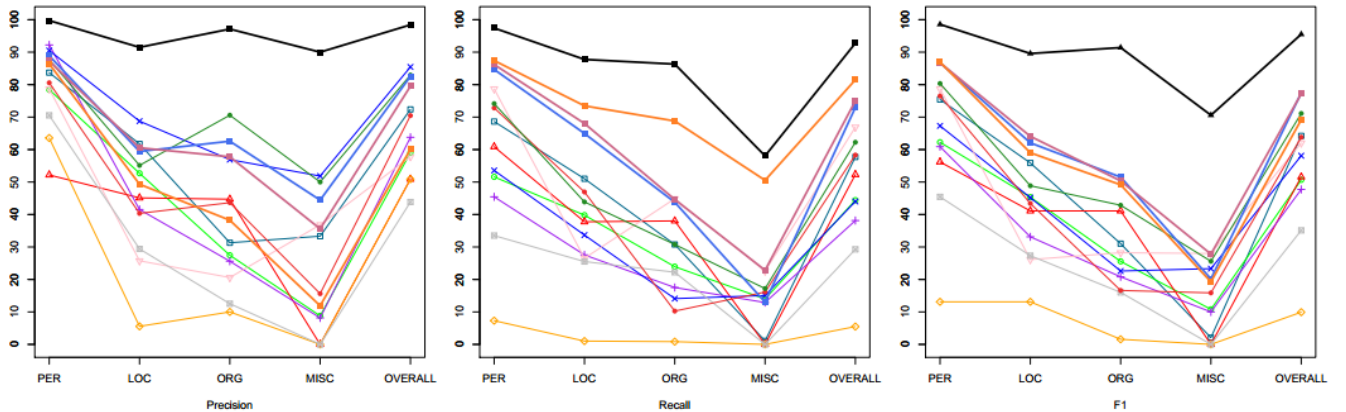
















Figure 2: Precision, Recall and F-measure results for individual NERD extractors, Stanford and NERD-ML on MSM2013 data set for different classes and overall. The black line denotes the upper limit the combined extractors can obtain.



Designation:

	AlchemyAPI
	DBpedia Spotlight
	Cicero
	Lupedia
	OpenCalais
	Saplo
	TextRazor
	Wikimeta
	Yahoo
	Stanford NER
	NERD-ML-NB
	NERD-ML-KNN
	NERD-ML-SVM
	Upper Limit

Results summary: Results show strengths and weaknesses of these linkers depending on the corpus. AlchemyAPI has generally the best precision, while dataTXT and TextRazor have the best recall when linking named entities to the normalized DBpedia knowledge base for respectively the newswire and the microposts corpora. Overall, TextRazor is the one, which shows the most stable and solid performance on both data sets when looking at the f-measure.

- Giuseppe Rizzo and Raphaël Troncy. ***NERD: Evaluating Named Entity Recognition Tools in the Web of Data***. In 10th International Semantic Web Conference (ISWC'11), Demo Session, Bonn, Germany, October 23-27, 2011
http://porto.polito.it/2440793/1/wekex2011_submission_6.pdf

Results summary:

Agreement Investigation Fleiss' Kappa score was computed to assess the agreement among the four raters. Low agreement level is obtained for the NE detection and its relevance for all extractors. Instead, an overall agreement is reached for AlchemyAPI, Extractiv and OpenCalais when users evaluated the Type and URI field. DBpediaSpotlight presents substantial agreement among all raters for the type field, instead low agreement for other fields due, essentially, to the heterogeneous results provided by the extractor (ie entity list includes named entities and often topic concepts affecting the overall evaluation). Instead, Zemanta shows an interesting agreement when URI field is evaluated.

Statistic Results. AlchemyAPI, although preserving good performance in NE extraction and accurate typing, has a clear weakness to link the NE to a web resource. URI disambiguation is better performed by Zemanta and DBpedia Spotlight. Moreover, Zemanta has a good reliability to recognize NE in contrast to DBpedia Spotlight. However, both lack the rich type classification. For what concerns DBpedia Spotlight, this result contrasts with the large ontology used to classify the extracted NEs. OpenCalais and Extractiv demonstrate good results in the type identification task.

- Collection of links to publications about NERD - <http://nerd.eurecom.fr/publications>

Final verdict

NERD gives unified access to many annotators and so is a good choice for comparison of different annotators. If a best suitable annotator is selected then NERD gives no additional value. Related annotators have similar interfaces as NERD (all of them can be used as REST API) but many of them provide additional features (eg graphical user interface for testing). Referenced quality evaluations express that particular annotators provide clearly higher quality (AlchemyAPI, OpenCalais, Zemanta) and are reasonable to use by APE if 'built-in' dictionaries are sufficient. A referred comparison of NERD-supported and other annotators shows also that StanfordNER has similar or better quality than the best NERD-annotator AlchemyAPI.