MULTISENSOR

Mining and Understanding of multilinguaL contenT for Intelligent Sentiment Enriched coNtext and Social Oriented inteRpretation

FP7-610411

D4.4

Semantic integration framework

Dissemination level:	Public
Contractual date of delivery:	Month 32, 30/06/2016
Actual date of delivery:	Month 33, 08/07/2016
Workpackage:	WP4 Multidimensional Content Integration and Retrieval System
Task:	T4.2 Mapping discovery and validation T4.3 Content alignment and integration
Type:	Prototype
Approval Status:	Final Draft
Version:	1.0
Number of pages:	35
Filename:	D4.4_SemanticIntegrationFramework_2016-07-08_v1.0.pdf

Abstract

This Deliverable reports the final versions of the developed methodologies for ontology and content alignment. Regarding ontology alignment the Deliverable reports on the progress that has been accomplished regarding the implementation of extensions to the Alignment API. Regarding content alignment, D4.4 reports on the developments regarding the Entity Alignment service which has been deployed as part of the Content Extraction Pipeline, and also reports on the developments regarding the final version of the Content Alignment

Pipeline for assessing article similarities and contradictions.

The information in this document reflects only the author’s views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	15/05/2016	Initial draft and table of contents	C. Doulaverakis (CERTH)
0.2	12/06/2016	First version of D4.4	C. Doulaverakis, D. Liparas (CERTH)
0.3	18/06/2016	Review of CERTH contribution	S. Vrochidis (CERTH)
0.4	24/06/2015	Second iteration with contributions	C. Doulaverakis, D. Liparas, T. Mironidis (CERTH)
0.5	01/07/2016	Integrated document	I. Kompatsiaris (CERTH)
0.6	05/07/2016	Internal review	S. Dasiopoulou (UPF)
1.0	08/07/2016	Final version	C. Doulaverakis (CERTH)

Author list

Organization	Name	Contact Information
CERTH	Charalampos Doulaverakis	doulaver@iti.gr
CERTH	Dimitris Liparas	dliparas@iti.gr
CERTH	Theodoros Mironidis	mironidis@iti.gr
CERTH	Stefanos Vrochidis	stefanos@iti.gr
CERTH	Ioannis Kompatsiaris	ikom@iti.gr

Executive Summary

In Deliverable 4.4 the progress of Tasks 4.2 and 4.3 of WP4 for the development of Ontology Alignment and Content Alignment techniques is reported. Section 1 gives a brief overview of the timeline of the Tasks and the Deliverable, while Section 2 places the two Tasks and their development within the overall MULTISENSOR system. Section 3 presents the Entity Alignment module, which is used to detect and remove redundant and resolve contradictory content that has been extracted by content extraction services. Section 4 presents the final version of the Content Alignment Pipeline, which operates offline on the MULTISENSOR knowledge base and discovers similar and contradictory articles. Section 5 presents the progress of the ontology alignment method, where the developments and extensions to Alignment API are documented and also presents the steps taken for making the source code publicly available. Section 6 concludes the Deliverable.

Abbreviations and Acronyms

ASR	Automatic Speech Recognition
CAP	Content Alignment Pipeline
CEP	Content Extraction Pipeline
EA	Entity Alignment
EL	Entity Linking
IEEE	Institute of Electrical and Electronics Engineers
JPEG	Joint Photographic Experts Group
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
MPEG	Moving Picture Experts Group
NE	Named Entity
NER	Named Entity Recognition
OWL	Ontology Web Language
OWL-QL	Ontology Web Language Query Language
OWL-DL	Ontology Web Language Description Language
RDF	Resource Definition Framework
RSS	Really Simple Syndication
SWRL	Semantic Web Rule Language
SIMMO	Socially Interconnected MultiMedia Object
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

Table of Contents

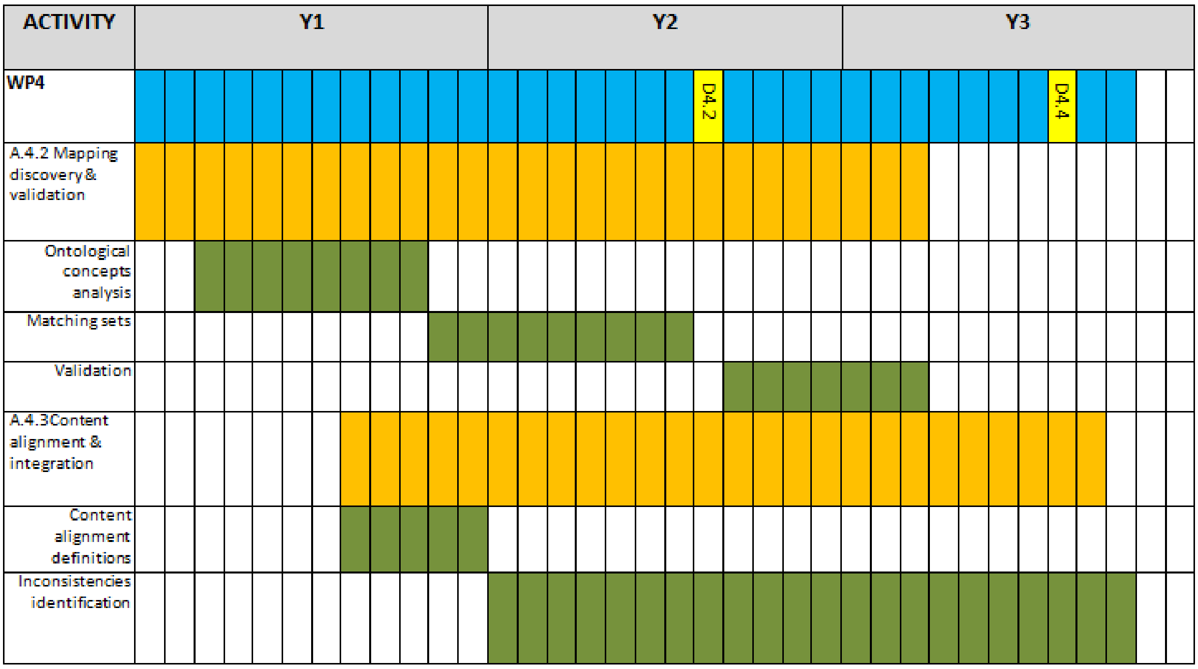
1 INTRODUCTION

This Deliverable reports on the progress of Tasks 4.2 “Mapping discovery and validation” and 4.3 “Content alignment and integration” of Work Package 4.

For T4.2, MULTISENSOR focuses on the development of new ontology alignment algorithms for concept matching, exploiting features that haven’t been considered so far in the ontology matching research area and on the development of an ontology alignment framework to support these activities in MULTISENSOR. D4.4 reports on the advancements that have been made from the previous reporting period and in particular from D4.2

For T4.3, the work carried out involves two directions that span beyond the reporting period of D4.2. The first direction involves the development of an entity alignment (EA) methodology for the discovery of contradictory or overlapping Entities in the content that is being produced by the various Content Extraction Pipeline (CEP) services. The second direction involves the development of a module for the discovery of related (similar or contradictory) articles within the knowledge base by taking into account the extracted RDF content. This module executes in a pipeline different from CEP and is termed Content Alignment Pipeline (CAP). CEP and CAP architecture is explained in detail in D7.2 and D7.4 where focus is given on their integration in the MULTISENSOR prototype.

The timeline of Tasks 4.2 and 4.3 along the project’s lifetime is the following (Figure 1):



ACTIVITY	Y1												Y2												Y3											
WP4	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	D4.2	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	D4.4	🟦	🟦		
A.4.2 Mapping discovery & validation	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧									
Ontological concepts analysis			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩																										
Matching sets											🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩																	
Validation																					🟩	🟩	🟩	🟩	🟩	🟩	🟩									
A.4.3Content alignment & integration								🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧			
Content alignment definitions								🟩	🟩	🟩	🟩	🟩																								
Inconsistencies identification													🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		

Figure 1: Timeline of Tasks 4.2 and 4.3

2 ARCHITECTURE

Section 2 gives an overview of the modules regarding their position in the MULTISENSOR processing pipeline and of a high level architecture of the modules that have been developed for T4.2 and T4.3 respectively.

2.1 Description

Ontology alignment (Task 4.2) is run as a separate processing module which is independently from the Content Extraction Pipeline (CEP) that has been described in Deliverable 7.2. Ontology alignment operates on the knowledge base of MULTISENSOR in order to identify equivalence relations (in ontological classes and properties) between the different ontologies of the knowledge base. While D4.2 gave a detailed description of the algorithmic developments of T4.2, in D4.4 we give more details on the implementation, the integration of these algorithms in an integrated graphical environment and the steps taken in order to make the source code publicly available.

Content alignment and integration (Task 4.3) involves a) the alignment of entities that are extracted from different services of the CEP so that contradictory information is identified and resolved, b) the Content Alignment Pipeline (CAP). The CAP involves the discovery of links between content items (articles) that are stored in the knowledge base. CAP is implemented as an offline method that executes in fixed intervals in order to identify these relations between content. As such it is executed in a different processing pipeline from CEP. The main differences of the currently developed version of CAP with the one reported in D4.2 is that this version requires no user input and it executes offline.

2.2 WP4 modules and pipelines

The module and the pipeline that have been described in D7.2 and are the subject of this deliverable are the “Mapping discovery and validation” module, and the “Content alignment pipeline”.

The “Mapping discovery and validation” module is independent of the CEP and runs separately as a standalone application. It takes as input two ontologies, usually these ontologies describe the same or similar domains, and produces a set of proposed mappings between the ontologies classes and properties. The main work that has been carried out involves the development of a user interface for ontology alignment based on open source software and the steps taken in order to distribute this tool as an open source framework. The process of ontology alignment is displayed in Figure 2.

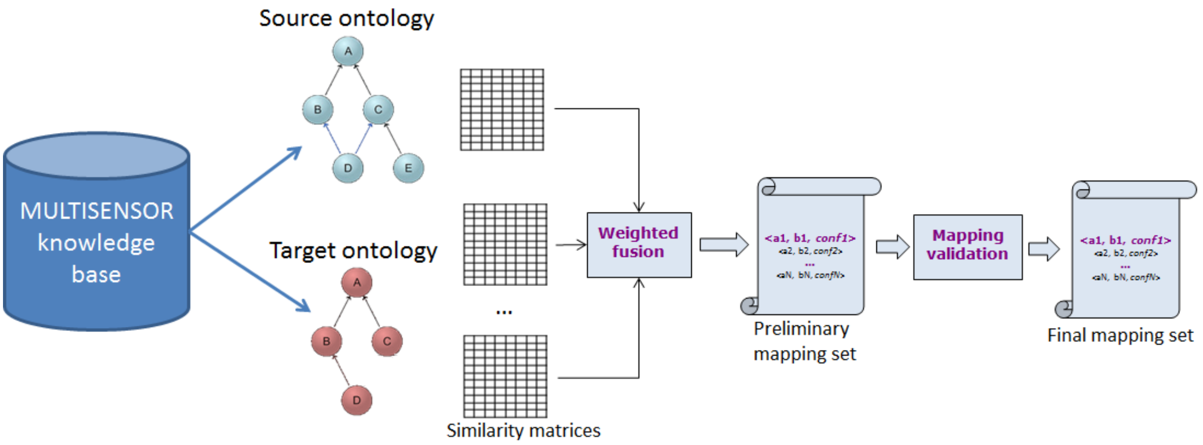


Figure 2: Basic workflow for aligning a source to a target ontology

For “Content alignment and integration” two different activities have taken place, a) the “Entity Alignment” (EA) module which processes the output of different content extraction services in CEP in order to remove redundant and resolve contradicting information within a content item (article) (Figure 3) and, b) the “Content alignment pipeline” (CAP) that performs on the knowledge base in order to discover relations (similarities and contradictions) between content items. Related content items are defined based on features such as Framenet graph similarity, named entity matching, matching of extracted concepts or spatiotemporal similarities. CAP is running independently from CEP and is a separate analysis process. It should be noted that the Description of Work, Task 4.3 is stated that it will deal with *“… a framework for integration/enhancement of overlapping/ complementary/possibly conflicting textual & visual descriptions”*, however textual descriptions only are considered for the current developments. The main reason for such a decision has to do with the fact that the “Content alignment and integration” modules are operating in the information that is stored in the MULTISENOR knowledge base which stores textual features and the results of other analysis modules.

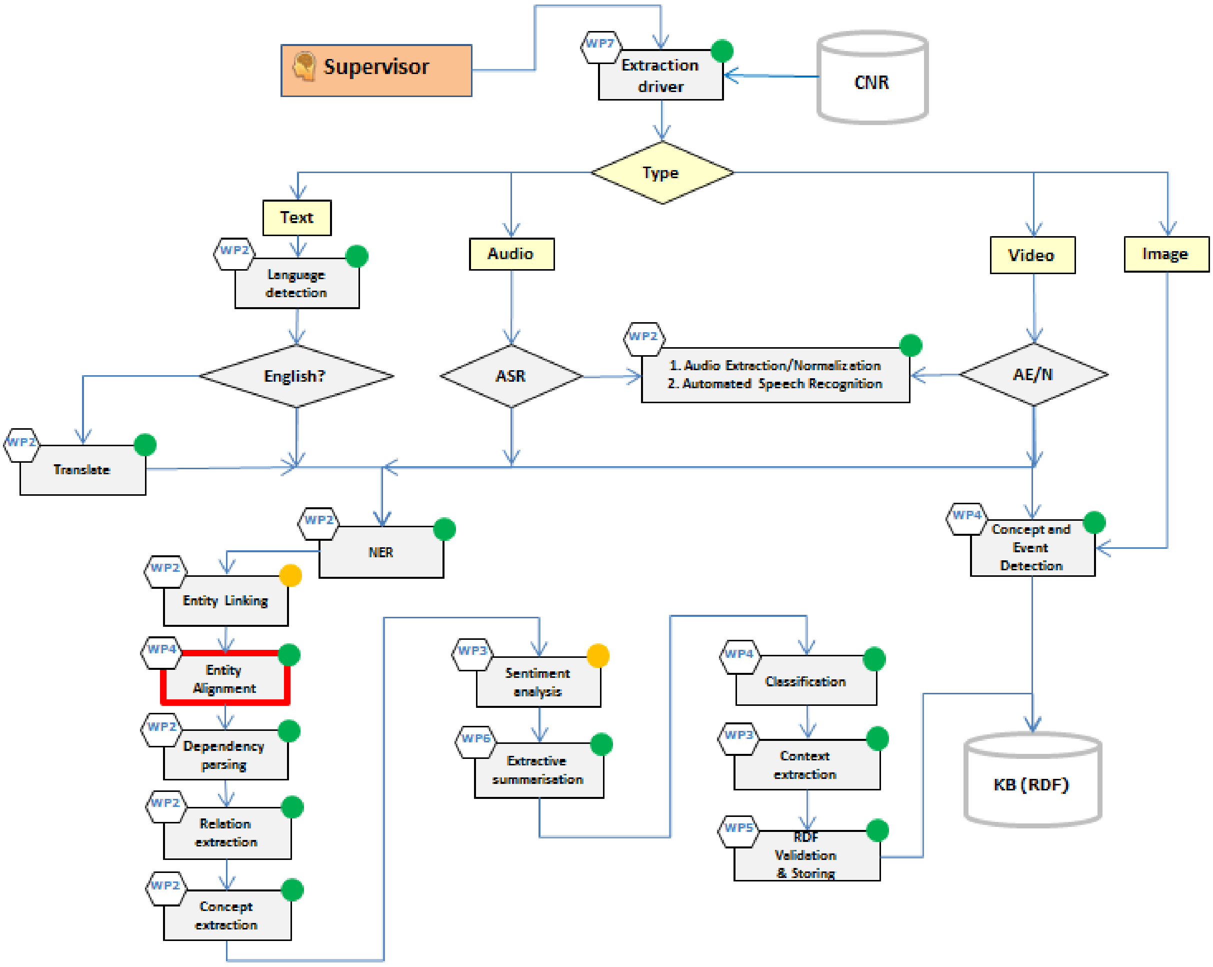


Figure 3: Position of the “Entity Alignment” module in the CEP (shown with thick red borders)

3 ENTITY ALIGNMENT

The Entity Alignment module is running as part of the CEP in the MULTISENSOR prototype. It is developed as part of the work for Task 4.3 "Content alignment and integration". Its purpose is to identify contradictory content that is extracted from a number of different services in the CEP and remove it, thus "cleaning" the knowledge base. This process results in a) keeping the knowledge base smaller in size as fewer triples are eventually stored and, b) most importantly, as the erroneous content is removed in the early stages of the processing pipeline, the subsequent modules that are running are able to produce more accurate results as is the case for example for the Deep Dependency parsing module.

The Entity Alignment module is run after the execution of the Named Entity Recognition (NER) and Entity Linking (EL) modules. In brief, these modules produce the following output.

- Named Entity Recognition: Scans the article text and identifies Named Entities (NE), e.g. persons, locations, organisations, amounts, timestamps, etc. The detected entities are additionally linked to DBpedia[1] entries or article-specific URI that correspond to these entities are created. Examples of the above are:
    - ***http://dbpedia.org/resource/Europe*** (DBpedia URI)
    - ***http://data.multisensorproject.eu/content/70cd1550bb04651603981f7cfcb58926b74444cd#Person=Gavin_Partington*** (MULTISENSOR URI)
- Entity Linking: In one of its functionalities, it scans the article texts and identifies named entities, such as persons or locations, and links these with Babelnet[2] ids. In addition, Babelnet is linked to DBpedia allowing facilitating the link of MULTISENSOR content to the linked data cloud. An example is:
    - ***http://babelnet.org/rdf/page/s00008659n*** (BabelNet URI) which is linked to DBpedia through the property *skos:exactMatch dbpedia:Margaret_Thatcher*

There are cases where the two services will identify the same NE but with different reference URIs thus ending up with different references to entities although they should be the same. This most frequently happens in multi-word expressions, such as "United States of America", where one service might recognize "United States", while the second one might recognize the whole expression "United States of America". Entity Alignment takes corrective actions to overcome this problem with post processing.

3.1 Related work in content alignment in knowledge bases

The emergence of Linked Data represents an important milestone for enabling large scale information sharing (Bizer et al., 2009). Linked Data can be created by exposing information that is coming from structured, semi-structured or unstructured sources. Data coming from semi-structured or unstructured sources is likely to contain erroneous RDF statements. Various techniques for improving the quality of data in Linked Data and knowledge bases have been presented and documented in literature. (Ji and Grishman, 2011) give an overview of the systems that took part in the Knowledge Base Population track at the 2010

[1] DBpedia, **http://wiki.dbpedia.org/**

[2] BabelNet, **http://babelnet.org/**

Text Analysis Conference. They also identify research challenges for future systems; among which issues related to the wider use of external knowledge bases and the need for context inference in order to link data with meaningful information. In (Paulheim and Bizer, 2013) the authors proposed a method for inferring the types for untyped resources which was based on the usage of information from the statistical distribution of properties and types. They assign probability values to subjects and objects that appear with predefined properties, e.g. *dbpedia-owl:location*, based on existing statements and determine the type of these untyped resources using these probabilities. They also proposed an approach for validating existing statements (Paulheim and Bizer, 2014) by assigning a confidence score to each statement which reflects the deviation of the types predicted by the statement from the statement's object's actual types. In (Knuth et al., 2012) while the authors do not present a methodology for detecting errors in knowledge bases, they propose an approach for collaboratively issuing correction requests of Linked Data to their publishers. They propose PATCHR as an ontology for enabling this methodology; however, they don't provide results on its usage and its effectiveness.

Applying alignment and removal of noise techniques on knowledge bases is mainly a data driven approach where one has to identify what type of noisy data are present in the knowledge base and based on that, design a strategy to apply corrective actions. This means that there is no single methodology for noise removal in knowledge bases but in most cases, heuristic methods are applied for solving such problems.

3.2 Process of aligning content in MULTISENSOR

3.2.1 Detection and linking of entities

As has been described earlier, the Named Entity Recognition (NER) and the Entity Linking (EL) services enable to scan the text for NEs and produce respective RDF descriptions. The NER service produces RDF triples that specify the NE type and its corresponding DBpedia URI for the Named Entities it identifies. The EL service accordingly produces RDF triples that link the detected Named Entities with BabelNet. An example of the output produced by the NER service is found below:

```
PREFIX ms: <http://data.multisensorproject.eu/content/>
PREFIX nerd: <http://nerd.eurecom.fr/ontology#>
PREFIX its: <http://www.w3.org/2005/11/its/rdf#>
PREFIX dbr: <http://dbpedia.org/resource/>

ms:05b91a2ba5d766f2e361bdaaccb0b925e395d092#char=3389,3396
its:taClassRef nerd:Location

ms:05b91a2ba5d766f2e361bdaaccb0b925e395d092#char=3389,3396
its:taIdentRef dbr:Atlanta
```

where it's identified that the entity between the indexes 3389 and 3396 is identified as a location, through the *its:taClassRef* property, and it's corresponding DBpedia URI has been given as *dbr:Atlanta* through the *its:taIdentRef* property.

In a similar fashion, the entities that are identified by the EL service will produce triples that are similar to

```
PREFIX ms: <http://data.multisensorproject.eu/content/>
PREFIX nerd: <http://babelnet.org/rdf/>
PREFIX its: <http://www.w3.org/2005/11/its/rdf#>

ms:05b91a2ba5d766f2e361bdaaccb0b925e395d092#char=1901,1915
its:taIdentRef bn:s02782809n
```

where the entity between the indexes *1901* and *1915* is linked to the BabelNet synset with id *s02782809n*.

The purpose of having such rich annotations is to be able to take advantage of Linked Data sources so that information regarding these annotations can be retrieved. For example, for Atlanta we can also get information from DBpedia regarding location coordinates

```
PREFIX wgs: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dbr: <http://dbpedia.org/resource/>

dbr:Atlanta wgs:lat "33.755"^^xsd:float
dbr:Atlanta wgs:long "-84.39"^^xsd:float
```

identify that it is a city

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>

dbr:Atlanta dbo:type dbr:City
```

or extract information regarding the country that it's located in through the *dbo:isPartOf* property chain

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>

dbr:Atlanta dbo:isPartOf dbr:Georgia_(U.S._state)
```

In addition, the entities that have been identified will be considered for further processing down the CEP from other services, such as the Dependency Parsing service.

3.2.2 Overlapping Named Entities

However, there are cases where the NER and EL services produce different annotations for the same Named Entity. This issue has been observed in situations where the NE to be annotated is a multi-word expression, i.e. it consists of more than one word, where one service would annotate a part of the whole word and the other service would annotate another part of the word, often assigning wrong links to DBpedia or BabelNet. Thus, the produced annotations indicate that these are different entities although they should be the same. An example of such an issue, taken directly from the CEP is illustrated below

```
PREFIX ms: <http://data.multisensorproject.eu/content/>
PREFIX nerd: <http://nerd.eurecom.fr/ontology#>
PREFIX its: <http://www.w3.org/2005/11/its/rdf#>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-
core#>

ms:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=1444,1461
its:taIdentRef dbr:Margaret_Thatcher ;
nif:anchorOf "Margaret Thatcher" .

ms:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=1453,1461
its:taIdentRef bn:s00076775n ;
nif:anchorOf "Thatcher" .
```

It can be observed from this example that while the first annotation was correctly linked to the DBpedia entry for Margaret Thatcher, the former UK Prime Minister, the second annotation missed the whole name and linked only the word "thatcher" to the BabelNet synset with id *00076775n* (thatcher: "Someone skilled in making a roof from plant stalks or foliage" which is obviously wrong. This annotation however was further propagated in the CEP and was processed by the dependency parsing service to produce further annotations such as

```
PREFIX ms: <http://data.multisensorproject.eu/content/>
PREFIX upf-deep: <http://taln.upf.edu/upf-deep#>

ms:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=1453,1461
upf-deep:deepDependency
ms:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=1439,1443
```

while additionally, the summarization services use this output to produce the article summaries. So it's clear that situations like the one above should be identified and corrected as they will be the cause of more errors further below in the pipeline, as concepts that should be part of multi-word expressions are considered as distinct entities.

3.2.3 Entity Alignment service in CEP

The Entity Alignment service that has been developed as part of the CEP is depicted as a block diagram in Figure 4.

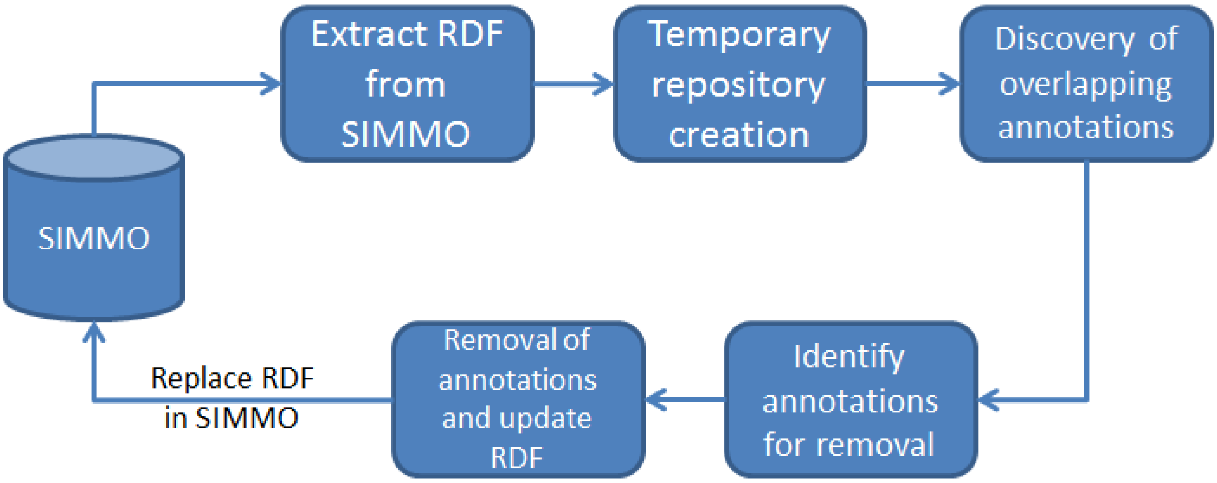

Extract RDF from SIMMO
Temporary repository creation
Discovery of overlapping annotations
SIMMO
Identify annotations for removal
Removal of annotations and update RDF
Replace RDF in SIMMO

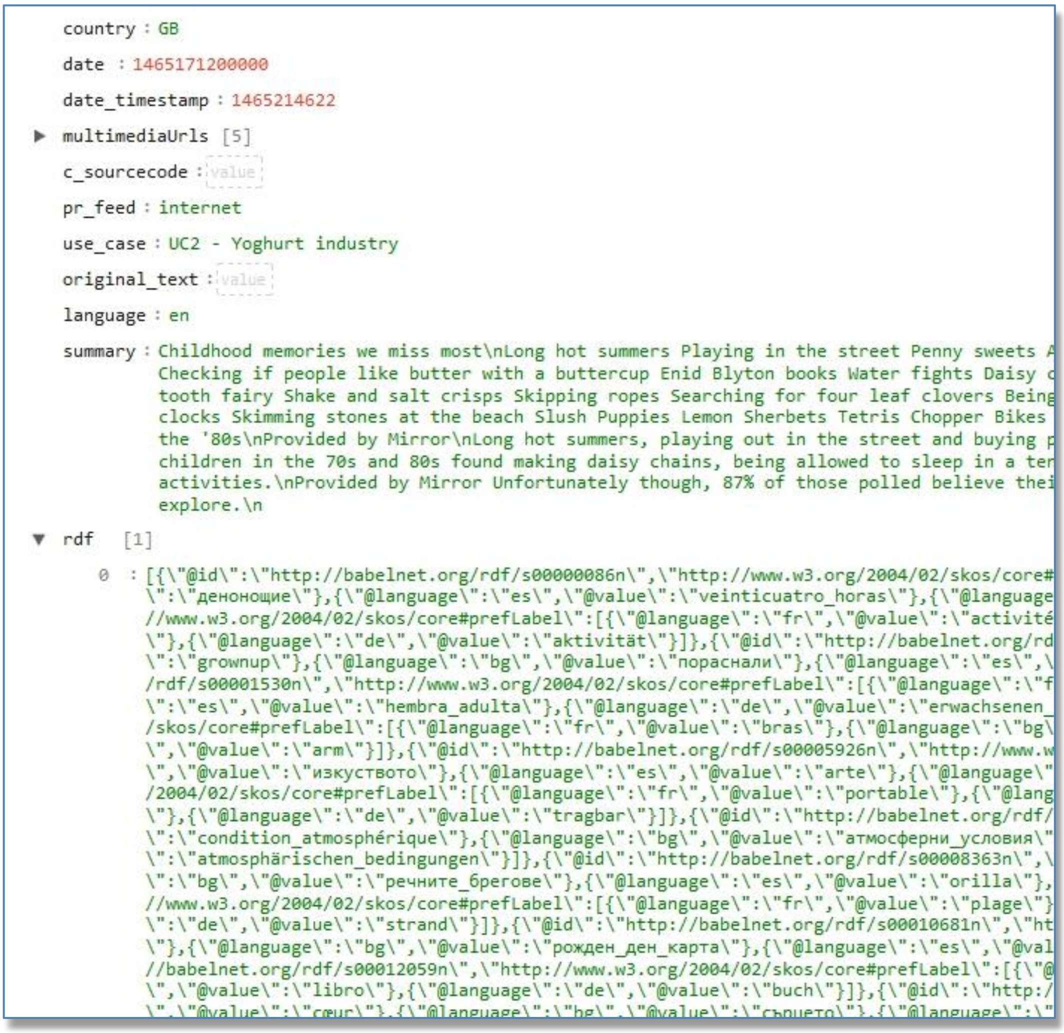
Figure 4: Overview of the Entity Alignment process

The process starts by taking the SIMMO that is passed within the CEP and extracting the RDF content for further processing. A sample of the SIMMO and it's RDF content is shown below, in Figure 5. The fields of the JSON string are visible along with their values. The RDF part corresponds to the "'*rdf*" field and it is encoded in JSON-LD[3], a rather complicated format as is illustrated, that makes working directly on the JSON-LD part impractical. In any case, JSON-LD was developed mainly for usage in web services where data are machine-processed and will not be directly used by humans.

The "*rdf*" portion of the SIMMO holds all the information that is generated by the different services of the CEP and Entity Alignment works on that portion of SIMMO. After the extraction of the RDF content, the next step is to create a temporary in-memory RDF repository where the data will be stored to. This is done for a variety of reasons:

a. Directly working on the JSON-LD string is not recommended as it is highly complex
b. Several APIs have been developed for creating RDF repositories and include functionalities for easily adding and deleting content. These APIs can handle JSON-LD both as input and as output (export the repository in JSON-LD format)
c. These repositories allow the execution of SPARQL queries, which is particularly convenient for the purposes of Entity Alignment as is explained further

[3] Overview of JSON-LD, **https://www.w3.org/TR/json-ld/**



```
   country : GB
   date : 1465171200000
   date_timestamp : 1465214622
▶  multimediaUrls [5]
   c_sourcecode : value
   pr_feed : internet
   use_case : UC2 - Yoghurt industry
   original_text : value
   language : en
   summary : Childhood memories we miss most\nLong hot summers Playing in the street Penny sweets A
             Checking if people like butter with a buttercup Enid Blyton books Water fights Daisy c
             tooth fairy Shake and salt crisps Skipping ropes Searching for four leaf clovers Being
             clocks Skimming stones at the beach Slush Puppies Lemon Sherbets Tetris Chopper Bikes
             the '80s\nProvided by Mirror\nLong hot summers, playing out in the street and buying p
             children in the 70s and 80s found making daisy chains, being allowed to sleep in a ten
             activities.\nProvided by Mirror Unfortunately though, 87% of those polled believe thei
             explore.\n
▼  rdf   [1]
      0  : [{\"@id\":\"http://babelnet.org/rdf/s00000086n\",\"http://www.w3.org/2004/02/skos/core#
           \":\"денонощие\"},{\"@language\":\"es\",\"@value\":\"veinticuatro_horas\"},{\"@language
           //www.w3.org/2004/02/skos/core#prefLabel\":[{\"@language\":\"fr\",\"@value\":\"activité
           \"},{\"@language\":\"de\",\"@value\":\"aktivität\"}]},{\"@id\":\"http://babelnet.org/rd
           \":\"grownup\"},{\"@language\":\"bg\",\"@value\":\"порасналм\"},{\"@language\":\"es\",\
           /rdf/s00001530n\",\"http://www.w3.org/2004/02/skos/core#prefLabel\":[{\"@language\":\"f
           \":\"es\",\"@value\":\"hembra_adulta\"},{\"@language\":\"de\",\"@value\":\"erwachsenen_
           /skos/core#prefLabel\":[{\"@language\":\"fr\",\"@value\":\"bras\"},{\"@language\":\"bg\
           \",\"@value\":\"arm\"}]},{\"@id\":\"http://babelnet.org/rdf/s00005926n\",\"http://www.w
           \",\"@value\":\"изкуството\"},{\"@language\":\"es\",\"@value\":\"arte\"},{\"@language\"
           /2004/02/skos/core#prefLabel\":[{\"@language\":\"fr\",\"@value\":\"portable\"},{\"@lang
           \"},{\"@language\":\"de\",\"@value\":\"tragbar\"}]},{\"@id\":\"http://babelnet.org/rdf/
           \":\"condition_atmosphérique\"},{\"@language\":\"bg\",\"@value\":\"атмосферни_условия\"
           \":\"atmosphärischen_bedingungen\"}]},{\"@id\":\"http://babelnet.org/rdf/s00008363n\",\
           \":\"bg\",\"@value\":\"речните_брегове\"},{\"@language\":\"es\",\"@value\":\"orilla\"},
           //www.w3.org/2004/02/skos/core#prefLabel\":[{\"@language\":\"fr\",\"@value\":\"plage\"}
           \":\"de\",\"@value\":\"strand\"}]},{\"@id\":\"http://babelnet.org/rdf/s00010681n\",\"ht
           \"},{\"@language\":\"bg\",\"@value\":\"рожден_ден_карта\"},{\"@language\":\"es\",\"@val
           //babelnet.org/rdf/s00012059n\",\"http://www.w3.org/2004/02/skos/core#prefLabel\":[{\"@
           \",\"@value\":\"libro\"},{\"@language\":\"de\",\"@value\":\"buch\"}]},{\"@id\":\"http:/
           \",\"@value\":\"cœur\"},{\"@language\":\"bg\",\"@value\":\"сърцето\"},{\"@language\":\"
```

Figure 5: Snapshot of a SIMMO (view from a JSON editor)

The RDF4J[4] repository API was selected (formerly known as Sesame API) as it is free and open source software which is mature and used in a number of projects in industry. After the extraction of the RDF content, this is imported in a memory model created with RDF4J.

Next step in the process is the discovery of annotations that are problematic. After an examination of the content that is extracted from the NER and EL services, it was identified that the problem of double annotations arises in cases of multiword expressions. In order to identify these expressions, the begin and end indices of the annotations were examined. In order to give an example of how the indices were used, the code snippet below gives an overview

```
PREFIX ms: <http://data.multisensorproject.eu/content/>
PREFIX nerd: <http://nerd.eurecom.fr/ontology#>
PREFIX its: <http://www.w3.org/2005/11/its/rdf#>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

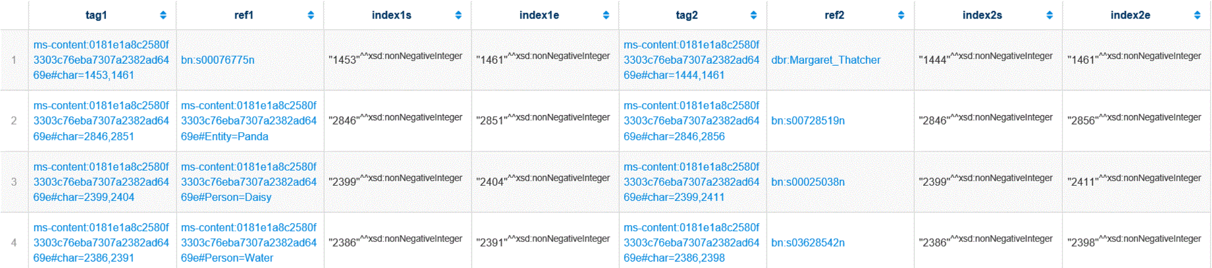
[4] RDF4J API, http://rdf4j.org/

```
PREFIX nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-
core#>

ms:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=1444,1461
its:taIdentRef dbr:Margaret_Thatcher ;
nif:anchorOf “Margaret Thatcher” ;
nif:beginIndex “1444”^^xsd:NonNegativeInteger ;
nif:endIndex “1461”^^xsd:NonNegativeInteger .

ms:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=1453,1461
its:taIdentRef bn:s00076775n ;
nif:anchorOf “Thatcher” ;
nif:beginIndex “1453”^^xsd:NonNegativeInteger ;
nif:endIndex “1461”^^xsd:NonNegativeInteger .
```

It can be observed that the string “Thatcher” of the second annotation, with begin and end indices 1453 and 1461 respectively, is included in the string “Margaret Thatcher” of the first annotation, with begin and end indices 1444 and 1461 respectively. As such, a way to identify these annotations is to check which are overlapping. The following SPARQL query does exactly this by comparing the indices of the annotations:

```
PREFIX its: <http://www.w3.org/2005/11/its/rdf#>
PREFIX nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-
core#>

SELECT * WHERE {
   ?tag1 its:taIdentRef ?ref1 .
   ?tag1 nif:beginIndex ?index1s .
   ?tag1 nif:endIndex ?index1e .
   ?tag2 its:taIdentRef ?ref2 .
   ?tag2 nif:beginIndex ?index2s .
   ?tag2 nif:endIndex ?index2e .
   FILTER (?index1s>=?index2s && ?index2e>=?index1e)
   FILTER (?tag1 != ?tag2)
}
```

A result of the execution of this query in a model is show in Figure 6 below.



	tag1	ref1	index1s	index1e	tag2	ref2	index2s	index2e
1	ms-content:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=1453,1461	bn:s00076775n	"1453"^^xsd:nonNegativeInteger	"1461"^^xsd:nonNegativeInteger	ms-content:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=1444,1461	dbr:Margaret_Thatcher	"1444"^^xsd:nonNegativeInteger	"1461"^^xsd:nonNegativeInteger
2	ms-content:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=2846,2851	ms-content:0181e1a8c2580f3303c76eba7307a2382ad6469e#Entity=Panda	"2846"^^xsd:nonNegativeInteger	"2851"^^xsd:nonNegativeInteger	ms-content:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=2846,2856	bn:s00728519n	"2846"^^xsd:nonNegativeInteger	"2856"^^xsd:nonNegativeInteger
3	ms-content:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=2399,2404	ms-content:0181e1a8c2580f3303c76eba7307a2382ad6469e#Person=Daisy	"2399"^^xsd:nonNegativeInteger	"2404"^^xsd:nonNegativeInteger	ms-content:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=2399,2411	bn:s00025038n	"2399"^^xsd:nonNegativeInteger	"2411"^^xsd:nonNegativeInteger
4	ms-content:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=2386,2391	ms-content:0181e1a8c2580f3303c76eba7307a2382ad6469e#Person=Water	"2386"^^xsd:nonNegativeInteger	"2391"^^xsd:nonNegativeInteger	ms-content:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=2386,2398	bn:s03628542n	"2386"^^xsd:nonNegativeInteger	"2398"^^xsd:nonNegativeInteger

Figure 6: Result of the SPARQL query execution for identifying overlapping annotations

After this step the process continues with selecting the annotations that will be removed. By performing a qualitative analysis, an examination of various results obtained by articles that had been processed, resulted to the conclusion that in the vast majority of the cases the annotation that correspond to the shortest string is the erroneous annotation. Using the API

for removal, all triples that refer to this annotation are removed. In the example of "Margaret Thatcher", the annotation
ms:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=1453,1461
will be removed as it annotates shorter text than
ms:0181e1a8c2580f3303c76eba7307a2382ad6469e#char=1444,1461

In the final step of the Entity Alignment process, after all erroneous annotations have been removed, the memory model is exported to JSON-LD and replaces the *"rdf"* entry in the SIMMO, and the latter is further propagated in the CEP to be consumed by subsequent services.

3.2.4 **Results**

The Entity Alignment service has been deployed successfully in the CEP. The average running time of the service for each article is around 1.8 seconds. The time is divided along the different processing steps as follows:

SIMMO parsing and model generation	Entity processing	Model serialization to JSON-LD and SIMMO update
920ms	420ms	361ms

Table 1: Average running times for various processing stages of Entity Alignment

In order to find out how often the Entity Alignment would identify contradicting entities, the CEP was monitored for a period of 5 days. During this period, 488 articles in total were processed and stored in the MULTISENSOR knowledge base (excluding tweets from Twitter.com as they carry less information due to their short length). Out of these, Entity Alignment had to be applied to 374 articles where on average 6.75 pairs of erroneous entities were detected.

3.2.5 **Source code availability**

The source code for the Entity Alignment service is uploaded in the SVN repository and can be found at
https://quark.everis.com/svn/MULTISENSOR/trunk/wp4/ms-svc-entityAlignment.

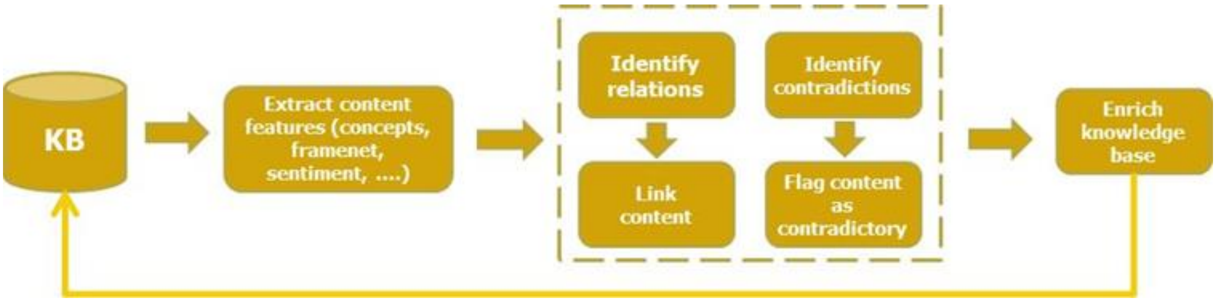
4 CONTENT ALIGNMENT PIPELINE

The content alignment pipeline (CAP) is a different processing flow from the CEP and is developed as part of the work for Task 4.3 "Content alignment and integration". While the CEP is running online for every article that is being retrieved from the various sources that are crawled by MULTISENSOR, the CAP is running offline at fixed intervals and performs across the knowledge base, finding relations between articles. Relations are considered either similarities or contradictions. A description of these relations is:

- Similarity: A number of articles are referring to the same subject, the same persons or locations or organizations, similar concepts or similar events are described that take place.
- Contradiction/opposing views: Articles that talk about the same subject use different expressions that result in different sentimentality or polarity.

Detecting article similarity has obvious advantages, e.g. finding out which articles talk about the same or similar subjects. Detecting contradictory article has the advantage of finding out articles that while they are referring to similar subjects some of their features are contradictory thus there is a potential to find disagreements between articles.

The workflow of the CAP is displayed in Figure 7.



KB
Extract content features (concepts, framenet, sentiment, ...)
Identify relations
Identify contradictions
Link content
Flag content as contradictory
Enrich knowledge base

Figure 7: Execution of CAP

The CAP process takes as input data retrieved from the knowledge base. It can be considered as a meta-process as it operates on information that has been extracted through processing of original data, the retrieved articles. The following sections describe in detail the relation measures that have been developed and how they are combined in order to assess the relation score between articles.

A number of novel measures for assessing the relation of content items based on the RDF content have been defined. The development and usage of these measures have been inspired by ontology alignment approaches where instead of assessing the similarity between ontological entities, the similarity between content items is calculated. Since there is a multitude of information extracted from different processing modules, primarily from the services integrated in the CEP, such as contextual information, named entities, concepts and FrameNet frames (Baker et al., 1998), the rationale is that for each information source different comparison methods should be applied for determining relation scores. At a final stage all relation scores are combined in order to end up with a single relation score for each pair of articles.

The description of Task 4.3 mentions that the methodology will be applied to features extracted from textual and audiovisual content. Audio content is considered for the purposes of Task 4.3 as the output of the Automatic Speech Recognition (ASR) module, which are the textual transcripts, are processed in the CEP and the results are integrated in the knowledge base as part of the article content. Regarding visual-based descriptions, multimodal content similarity is exploited for the purposes of Task 4.4, while the utilization of visual-based descriptions for identifying content contradiction would not be efficient so no effort was allocated to that direction.

4.1 Similarity measures for content alignment

The measures that have been developed to assess article similarity are described in detail in the following sections.

4.1.1 FrameNet-based similarity

FrameNet is an electronic resource based on the theory of frame semantics (Fillmore, 1982) and documents semantic and syntactic valences of words in their various senses (Ruppenhofer et al., 2006). In a sense it describes in a formal manner the events in a sentence, indicates who have participated in this event and other features such as temporal characteristics. The main concepts of FrameNet are:

- **Frames**: These are the conceptual structures that describe a particular type of situation, object or events, along with its participants and properties.
- **Frame Elements (FE)**: These are unique to each frame and denote the respective participants and what their role is.
- **Lexical Unit**: Words or phrases in the text that trigger a Frame.

An example of the usage of FrameNet with the sentence *“Electrolux announce today the theme of its design competition”* and how part of this sentence will be annotated by FrameNet is “*Electrolux announced today the theme …”*

Frame: *announced* (Statement)

FE: *Electrolux* (Speaker), *theme* (Message), *today* (Time)

The structure of the Frame and Frame Elements along with their relation is shown in Figure 8.

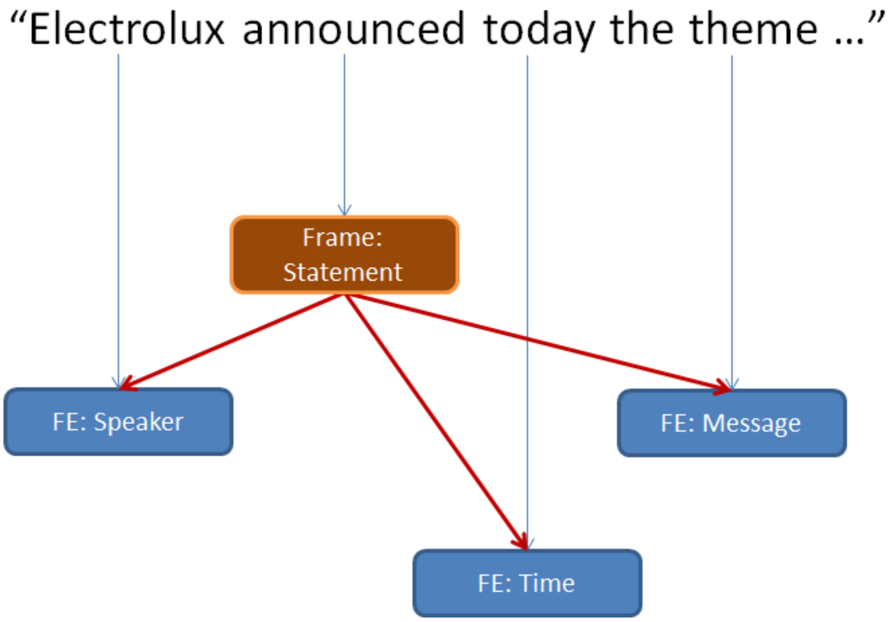


Figure 8: FrameNet structure for the example sentence shown above

Frames such as the above are extracted from different sentences in an article and these are encoded in RDF triples according to (Alexiev and Casamayor, 2016). An example of such a representation is shown here:

```
PREFIX ms: <http://data.multisensorproject.eu/content/>
PREFIX fn: <http://www.ontologydesignpatterns.org/ont/framenet/tbox/>
PREFIX lu: <http://www.ontologydesignpatterns.org/ont/framenet/abox/lu/>
PREFIX frame:
<http://www.ontologydesignpatterns.org/ont/framenet/abox/frame/>

ms:b8374d25b5c2ec25f2211855239807d4d39e8e22#char=180,189_annoSet_1 a
fn:AnnotationSet ;
    fn:annotationSetFrame frame:Statement ;
    fn:annotationSetLU lu:announce.v ;
    fn:hasLayer
ms:b8374d25b5c2ec25f2211855239807d4d39e8e22#char=180,189_layerFE_1 .

ms:b8374d25b5c2ec25f2211855239807d4d39e8e22#char=180,189_layerFE_1 a
fn:Layer ;
    fn:layer_name "FE" ;
    fn:hasLabel
ms:b8374d25b5c2ec25f2211855239807d4d39e8e22#char=169,179_fe_1 ,
ms:b8374d25b5c2ec25f2211855239807d4d39e8e22#char=190,195_fe_1 ,
ms:b8374d25b5c2ec25f2211855239807d4d39e8e22#char=200,205_fe_1 .
```

Frames in FrameNet are structured hierarchically and in addition FrameNet has been expressed as a RDF ontology[5]. An illustration of this ontology is shown in Figure 9 where the hierarchical structure of FrameNet is depicted.

[5] Framenet ontology, http://www.ontologydesignpatterns.org/ont/framenet/abox/cfn.rdf

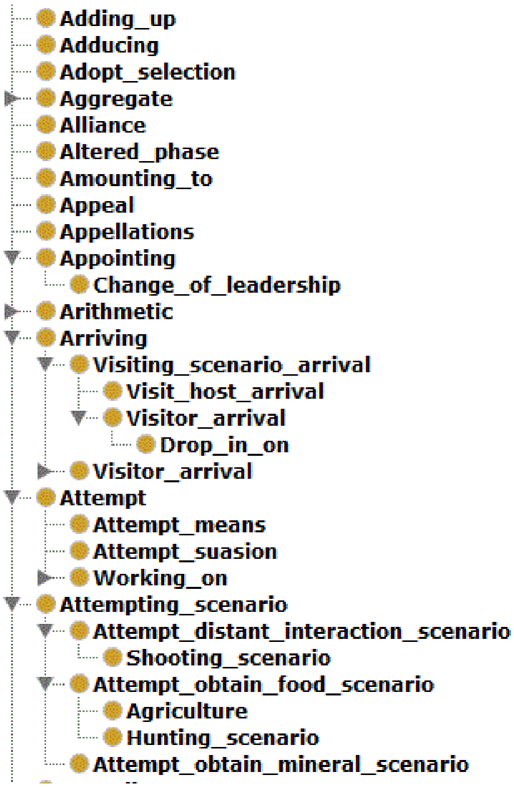


Figure 9: Illustration of the FrameNet ontology

For obtaining the Frames annotations of an article the following query is issued:

```
PREFIX rdf: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ms: <http://data.multisensorproject.eu/content/>
PREFIX fn: <http://www.ontologydesignpatterns.org/ont/framenet/tbox/>
PREFIX nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-
core#>

SELECT ?frames WHERE {
   ?root nif:oliaLink <http://taln.upf.edu/olia/penn-dep-syntax#root> .
   ?root nif:oliaLink ?anno .
   ?anno rdf:type fn:AnnotationSet .
   ?anno fn:annotationSetFrame ?frames .
}
```

At a second step, for every *?frame* that is retrieved using the query above, the following query retrieves the Frame Elements:

```
PREFIX rdf: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ms: <http://data.multisensorproject.eu/content/>
PREFIX fn: <http://www.ontologydesignpatterns.org/ont/framenet/tbox/>
PREFIX nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-
core#>

SELECT ?fe WHERE {
   ?root nif:oliaLink <http://taln.upf.edu/olia/penn-dep-syntax#root> .
   ?root nif:oliaLink ?anno .
   ?anno rdf:type fnt:AnnotationSet .
   ?anno fn:annotationSetFrame ?frame .
   ?anno fn:hasLayer ?layer .
   ?layer fn:hasLabel ?nodes .
   ?nodes fn:label_FE ?fe .
}
```

For every article a number of Frames and the related Frame Elements are extracted. Through the ontology of Figure 9 a graph can be constructed using the Frames and Frame Elements that have been detected. Using this graph representation, article similarity can be evaluated by comparing the graphs, by employing graph similarity metrics such as (Nikolic, 2012), as articles that have similar FrameNet graph representations will also have similar content. The similarity value $sim_{framenet}$ is calculated. While it's true that there are cases where lexical units with opposing meanings can evoke the same Frame, for example the lexical units *hate* and *like* both evoke the Frame *Experiencer_focus*, thus a Framenet annotation can be represented by the same graph but have different meaning, this metric focuses on the entity relation structure, i.e. that two entities are taking part in an action. Thus while the action might be different in meaning, the fact that are both referenced as interacting is an indication of similarity between articles.

4.1.2 Concept similarity

As part of the CEP, the Concept Extraction and Entity Linking services extract important concepts from an article. In other words, these concepts are the most prominent, content-wise, in the article and can be used to characterize it. There are two different types of concepts that are extracted:

a. Concepts that are generic and characterize an article. These are annotated using the triple *<concept> its:taClassRef mso:GenericContent*.
b. Concepts that are specific to the use cases and also characterize an article. These are annotated using the triple *<concept> its:taClassRef mso:SpecificContent*.

Concepts can be considered as a compact representation of an article's contents. Under this assumption, if two articles have common concepts then they can be considered similar. For retrieving an article's concepts, the following query is issued:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-
core#>
PREFIX its: <http://www.w3.org/2005/11/its/rdf#>
PREFIX mso: <http://data.multisensorproject.eu/ontology#>

SELECT ?root ?lemma ?anchor WHERE {
   #Generic concepts that characterize an article
   {  ?root its:taClassRef mso:GenericConcept .
      ?root nif:lemma ?lemma .
      OPTIONAL {?root nif:anchorOf ?anchor}
   }
   UNION
   #Specific concepts that are pertinent to the use cases
   {
      ?root its:taClassRef mso:SpecificConcept .
      ?root nif:lemma ?lemma .
      FILTER NOT EXISTS {?root its:taIdentRef ?ref} .
      OPTIONAL {?root nif:anchorOf ?anchor}
   }
}
```

The *?lemma* variable holds the concept word label. Comparing the lists of concepts between two articles is used to calculate their similarity as

$$sim_{concepts}(a,b) = \frac{common_{a,b}}{\min(L_a, L_b)}$$

where L_a, L_b are the lengths of the concept lists of the two articles *a*,*b*. $common_{a,b}$ is the number of common concepts between *a*,*b* such that

$$m_{a_i,b_j} = \begin{cases} 1, & a_i = b_j \\ 0, & a_i \neq b_j \end{cases} \text{ and}$$

$$common_{a,b} = \sum_{i=0}^{L_a} \sum_{j=0}^{L_b} m_{a_i,b_j}$$

A variation of the above measure compares the concepts not using exact match but by taking advantage of external linguistic resources, such as Wordnet or BabelNet, and establishing synonymy or hypernym/hyponym relations. m_{a_i,b_j} would then be expressed as

$$m_{a_i,b_j} = \begin{cases} 1 & a_i = b_j \\ 1 & a_i \ synonym \ b_j \\ 1/N & a_i \ hypernym \ or \ hyponym \ b_j \ with \ N \ hops \\ 0 & otherwise \end{cases}$$

4.1.3 Named Entity similarity

Named entities are extracted from articles using different modules, as has been explained in Section 3. For each article, the list of named entities can be retrieved using the following query

```
PREFIX nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-
core#>
PREFIX its: <http://www.w3.org/2005/11/its/rdf#>

SELECT ?link ?neType ?lemma WHERE {
   ?ne nif:lemma ?lemma .
   ?ne its:taClassRef ?neType .
   ?ne its:taIdentRef ?link .
}
```

As with concepts, two articles are considered similar if they are referring to the same named entities (places, locations, organizations). Using the measure

$$sim_{NEs}(a,b) = \frac{common_{a,b}}{\min(L_a, L_b)}$$

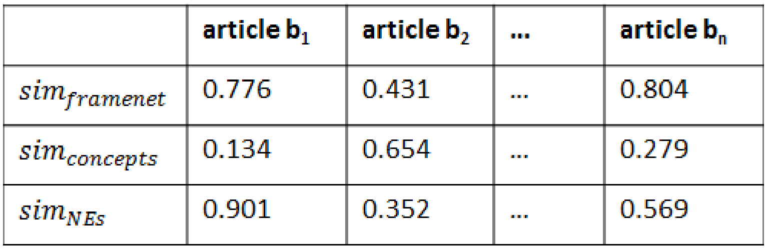
where $common_{a,b}$ is the number of common NEs between articles a,b and L_a, L_b are the lengths of the named entity lists for a,b respectively. The equality of NEs is determined by their URIs.

4.1.4 **Combination of similarity measures**

The purpose of the CAP is to compute relation values between articles in the knowledge base. As the knowledge base constantly expands and new articles are processed, it is not computationally feasible to compute the relation score of an article with all the other articles in the knowledge base[6]. In order to tackle this problem, each article is compared with 2000 other articles that are temporally proximal, i.e. the first 1000 articles with date before the article in question (older) and the first 1000 articles with date after the article (earlier). A comparison list is created. This allows keeping processing time at a reasonable amount.

Each article is compared to all other articles in the list using each of the measures described above, thus creating a vector of relation values for each measure. Each similarity measures described above, all produce a similarity value in the range of $0 \leq sim \leq 1$. An example of such a calculation is displayed in Figure 10 where an article *a* is compared to *n* other articles using the different measures and each one produces different results. This is an approach similar to ontology alignment. While the total similarity of *a* with *b*$_1$, *b*$_2$, …, *b*$_n$ can be computed by simply adding the different scores, this approach would have drawbacks as has been mentioned in D4.2. In short the main disadvantages of such an approach would be:

1. When comparing values different measures, the values of one measure might imply different relation strength than the same value of another measure
2. Different measures often display different distribution of values, e.g. one measure more frequently assigns values from 0.6 and up and another assigns values across the whole range of [0,1]



	article b$_1$	**article b$_2$**	**…**	**article b$_n$**
$sim_{framenet}$	0.776	0.431	…	0.804
$sim_{concepts}$	0.134	0.654	…	0.279
sim_{NEs}	0.901	0.352	…	0.569

Figure 10: Example of a run where an article *a* is compared to a number of other articles

The method that is used is the weighted fusion method that has been proposed in D4.2 for the combination of different ontology matchers

The score for each article pair is given by

$$S(a,b) = \sum_k w_{a,b} sim_k(a,b)$$

[6] At the time of writing, the MULTISENSOR knowledge base holds 85,308 articles

with

$$w_{a,b} = \frac{1}{1 + e^{\gamma * (s(a,b) - (\mu_a + \beta * \sigma_a))}}, \qquad with\ 0 < w_{a,b} \leq 1$$

where $w_{a,b}$ is the weight factor for *sim(a,b)*. *sim(a,b)* is the relation value of the *(a,b)* article pair, μ_a are the mean and standard deviation, respectively, of the relation values of all the pairs between source article a and all target articles. This weighting scheme promotes measures with high variance in their scores rather than measures that e.g. consistently assign high or low value scores thus are not as discriminative.

4.2 Contradiction measures for content alignment

Similarity between articles can be considered a measure that is easily interpreted and is common in the area of information retrieval. In MULTISENSOR one of the use cases is about journalism and showcases how the MULTISENSOR system can be exploited so that interesting information can be found more easily, compared to existing tools. For journalists, of particular interest is the discovery "contradicting" information since this enables them to have a view of different opinions regarding the same or similar subjects and thus enabling them to document the "whole picture" in a news story.

The approach that was followed was to develop a measure for assessing contradictory content in articles. Since contradiction makes sense between articles that are referring to similar events or news, the contradiction measures that are described below are applied to articles that have been found to be similar. The contradiction measures are applied to articles that have been identified as most similar by the similarity measures described in Section 4.1.

4.2.1 Exploiting sentiment analysis data

As part of the CEP, sentiment analysis services have been integrated that extract sentiment-related information. This analysis extracts sentimentality and polarity values that give indications of how objective and biased, respectively, an article is. As analysis results it can give interesting information regarding the content of an article if, for example, two articles talk about Germany's energy policy and one article has high sentimentality value, thus expressing positive sentiments, while another has low sentimentality value, thus expressing negative sentiments. From a user's perspective this would indicate an interesting result as two different opinions on a subject can be observed and documented.

For exploiting the sentimentality and polarity values, the following approach is followed:

For each article, an article similarity list L_s is generated, using the measures described in Section 4.1. As has been explained, contradiction in articles makes more sense if it is computed in articles with similar content, i.e. they are similar. For each of the articles in L_s their sentimentality and polarity values are retrieved from the knowledge base using the following query

```
PREFIX nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-
core#>
PREFIX mso: <http://data.multisensorproject.eu/ontology#>
PREFIX marl: <http://purl.org/marl/ns#>

SELECT * WHERE {
   ?context nif:sourceUrl ?url .
   ?context nif:opinion ?opinion .
   ?opinion mso:negativePolarityValue ?val1 .
   ?opinion mso:positivePolarityValue ?val2 .
   ?opinion mso:sentimentalityValue ?val3 .
   ?opinion marl:polarityValue ?val4 .
}
```

The *?polarityValue* is the sum of *?negativePolarityValue* and *?positivePolarityValue*. The sentiment score is given as

$$score_{a,b_{sentiment}} = \frac{|a_{sent} - b_{sent}| + |a_{pol} - b_{pol}|}{2 * 8}$$

where $a_{sent}, a_{pol}, b_{sent}, b_{pol}$ are the sentimentality and polarity values for articles a,b, with $0 \leq contr_{sentiment} \leq 1$. The sum of differences of polarity and sentimentality are divided by 8 to keep the score values in the space [0,1] since $-4 \leq polarity \leq 4$ and $0 \leq sentimentality \leq 8$.

The final contradiction score for sentiment-based contradiction is calculated as

$$contr_{a,b_{sentiment}} = sim_{a,b} * score_{a,b_{sentiment}}$$

where $sim_{a,b}$ is the similarity value between articles a,b which is retrieved from the list L_s.

4.3 RDF representation of CAP

Results of the analysis of the CAP are inserted in the knowledge base. One feature of the similarities that are computed is that they are symmetric, which means that if $sim_{a,b} = x$ then also $sim_{b,a} = x$. Taking this under consideration and that the similarities cannot be considered as part of the article analysis, the CAP results are stored in the knowledge base in a separate graph. An example of the graph contents is shown below:

```
PREFIX oa: <http://www.w3.org/ns/oa#>
PREFIX mso: http://data.multisensorproject.eu/ontology#
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

GRAPH <http://data.multisensorproject.eu/CAP> {
  <http://data.multisensorproject.eu/CAP/123> a oa:Annotation;
    oa:hasBody
<http://data.multisensorproject.eu/content/53a0938bc4770c6ba0e7d7b9ca88a637
f9e9c304>,
<http://data.multisensorproject.eu/content/ca34bb35770bfa55434a0689d64e1e6a
60611047>;
    mso:score 0.862;
    oa:motivatedBy mso:linking-similar;
    oa:annotatedBy <http://data.multisensorproject.eu/agent/CAP>;
    oa:annotatedAt "2016-01-11T12:00:00"^^xsd:dateTime .

  <http://data.multisensorproject.eu/CAP/124> a oa:Annotation;
    oa:hasBody
<http://data.multisensorproject.eu/content/53a0938bc4770c6ba0e7d7b9ca88a637
f9e9c304>,
<http://data.multisensorproject.eu/content/57e07befbda355c2eca2ee521926071e
e9f5c719>;
    mso:score 0.795;
    oa:motivatedBy mso:linking-contradictory;
    oa:annotatedBy <http://data.multisensorproject.eu/agent/CAP>;
    oa:annotatedAt "2016-01-12T12:00:00"^^xsd:dateTime .
}
```

4.4 Source code availability

The source code for the CAP service is uploaded in the SVN repository and can be found at:

https://quark.everis.com/svn/MULTISENSOR/trunk/wp4/ms-svc-contentAlignment.

4.5 Evaluation

The CAP was tested in a subset of the MULTISENSOR dataset. A total of 100 articles were randomly retrieved from the MULTISENSOR knowledge base. Similar and contradictory articles were manually labelled. A total of 7 categories were used to manually categorize the articles. These categories and the number of respective articles are shown in Table 2. Note that some articles were categorized in more than one category, e.g. Politics and Economics. All articles in a category are labelled as similar. This categorization is the ground truth set for similarity.

Category	Number of articles
Crime reports	46
Economy	9
Tourism	9
Food	18
Politics	12
Health	7
Stories	23

Table 2: Article categorization

For performing the evaluation, a local repository was created and the CAP process was applied on it. The outcome was that for each article two lists of similar and contradictory articles were generated. Precision was calculated for the top 7 articles of each list (as the category with the least amount of articles is Health with 7 articles). The Mean Average Precision was calculated for all 100 articles.

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

and

$$Mean\ Average\ Precision\ (MAP) = \frac{\sum_{i=1}^{N} Precision_i}{N}$$

Where N $(N=100)$ is the number of articles and $Precision_i$ is the precision that was calculated for article i.

For similarity, in order to examine how well the different measures that have been defined, are performing, their results are displayed in Table 3. It should be noted that the baseline approach is the Named Entity similarity measure, as named entities were utilized for the development of the baseline CAP version, reported and described in D4.2. Compared to the indicators established in D1.2, a greater than the expected 5% improvement over the baseline is achieved by the combination of all similarity measures (see Table 4).

	MAP
Named Entity similarity	0.548
FrameNet similarity	0.420
FrameNet + Named Entity	0.602
Concept similarity	0.672
Named Entity + FrameNet + Concept similarity	**0.682**

Table 3: MAP for similarity measures

	Named Entity similarity (baseline)	Named Entity + Concept + Framenet similarity
Precision	0.540	0.682

Table 4: Comparison of proposed method (combination of all similarity measures) against baseline approach

For evaluating the sentiment-based contradiction measure, the articles of each category were examined and a list of contradictory articles was defined for each article based on the perceived negative or positive feeling that the articles were carrying. Articles are labelled as contradictory or not. The sentiment-based contradiction results were obtained as presented in Table 5 (MAP was calculated).

	Sentiment-based contradiction
MAP	0.579

Table 5: MAP for sentiment-based contradiction

5 ONTOLOGY ALIGNMENT

This section describes the advances that have been made regarding Task 4.2 “Mapping discovery and validation”. In D4.2, a detailed description of the algorithmic developments for ontology alignment was given. D4.4 focuses on the developments for the actual software of ontology alignment, the extensions that were made in the Alignment API v4.6[7] in order to support the ontology alignment in the MULTISENSOR ontology alignment Task and the steps taken in order to publish these extensions as open source code.

5.1 Enabling the aggregation of matching algorithms in Alignment API

Alignment API is a popular framework for developing ontology alignment algorithms (Euzenat, 2004). It integrates methods for accessing ontologies, running the alignment algorithms on selected resources (classes, properties or instances) and retrieving results. It is distributed as open source software under the license GNU LGPL[8]. This allows the source code to be used freely and modified.

However, in order to be able to use it for the purposes of MULTISENSOR ontology alignment processes several adaptations had to be made to the handling and execution of the alignment algorithms. The most notable adaptation was the way that results from multiple alignment algorithms are aggregated in order to end up with a single alignment list between ontologies. Alignment API uses implements the class *ConcensusAggregator* for combining the results of different algorithms, although it adopts a simplistic approach where all scores are added and if the result is larger than a supplied threshold, then the match is considered valid. The use of the weighted fusion approach that was described in D4.2 requires more advanced manipulation of the different algorithms results.

For implementing this approach, the internal structure of how Alignment API handles similarity scores between ontology resources was changed. Originally, the class *BasicCell* corresponds to an alignment pair. *BasicCell* members are displayed below

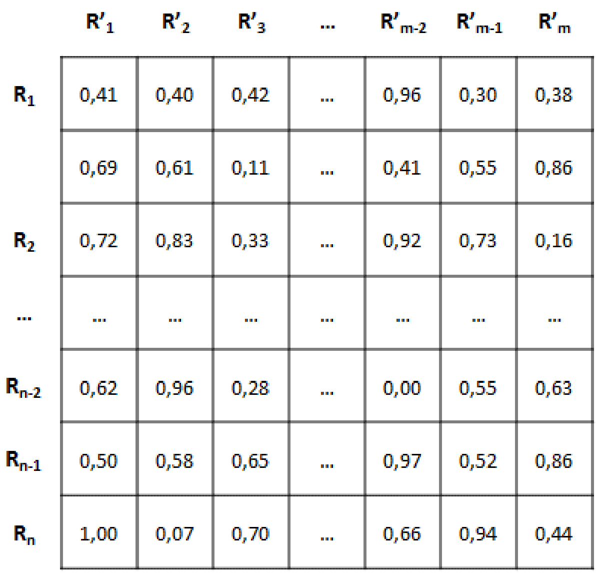
```
public class BasicCell implements Cell, Comparable<Cell> {

    protected String id = null;
    protected String semantics = null;
    protected Object object1 = null;
    protected Object object2 = null;
    protected Relation relation = null;
    protected double strength = 0;
    protected Extensions extensions = null;
```

An alignment is a collection of *BasicCell* objects. While this approach will work for a single algorithm or for simple operations between algorithm results, the implementation of weighted fusion that was described in D4.2 would be sub-optimal as it is not possible to easily access the mapping score results of one resource to all others thus an iteration over all *BasicCell* objects would be required to get the required alignment pairs. For allowing an

[7] Alignment API, http://alignapi.gforge.inria.fr/

[8] GNU Lesser General Public License (LGPL), https://www.gnu.org/licenses/lgpl-3.0.en.html

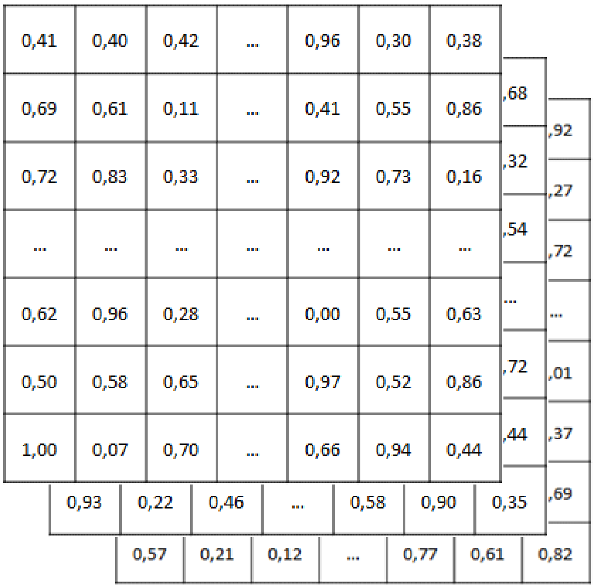
efficient access to alignment pairs, in MULTISENSOR all computed scores are stored in a similarity matrix. Given two ontologies O_1 and O_2, where O_1 contains *n* ontological resources (classes, properties or instances) and O_2 *m* resources, the dimensions of the matrix *M* are *n,m.* Each cell *(i,j)* of *M* contains the similarity score between the i_{th} resource of O_1 and the j_{th} resource of O_2. This is illustrated in Figure 11.



	R'1	R'2	R'3	…	R'm-2	R'm-1	R'm
R1	0,41	0,40	0,42	…	0,96	0,30	0,38
	0,69	0,61	0,11	…	0,41	0,55	0,86
R2	0,72	0,83	0,33	…	0,92	0,73	0,16
…	…	…	…	…	…	…	…
Rn-2	0,62	0,96	0,28	…	0,00	0,55	0,63
Rn-1	0,50	0,58	0,65	…	0,97	0,52	0,86
Rn	1,00	0,07	0,70	…	0,66	0,94	0,44

Figure 11: Example of a similarity matrix

Using this alignment pair structure, accessing alignments of a resource R_i with resources R'_j is easy and efficient. Each alignment algorithm produces stores its alignment results (scores) in such a similarity matrix and eventually after all alignment algorithms have executed, we end up with an equal number of similarity matrices (Figure 12).



0,41	0,40	0,42	…	0,96	0,30	0,38
0,69	0,61	0,11	…	0,41	0,55	0,86
0,72	0,83	0,33	…	0,92	0,73	0,16
…	…	…	…	…	…	…
0,62	0,96	0,28	…	0,00	0,55	0,63
0,50	0,58	0,65	…	0,97	0,52	0,86
1,00	0,07	0,70	…	0,66	0,94	0,44

Figure 12: List of similarity matrices

Using such an approach, the all corresponding similarity pairs in each algorithm are accessible in the same way, by referencing the indices *(i,j)* of each matrix, thus allowing an efficient access to all alignment results.

5.2 Results

The results displayed here are repeated from D4.2 where the approach of the weighted fusion method is benchmarked against the OAEI 2012 bibliographic dataset. Baseline F-measure value is 0.68 while the weighted fusion baseline value is 0.73, where F-measure s defined as:

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

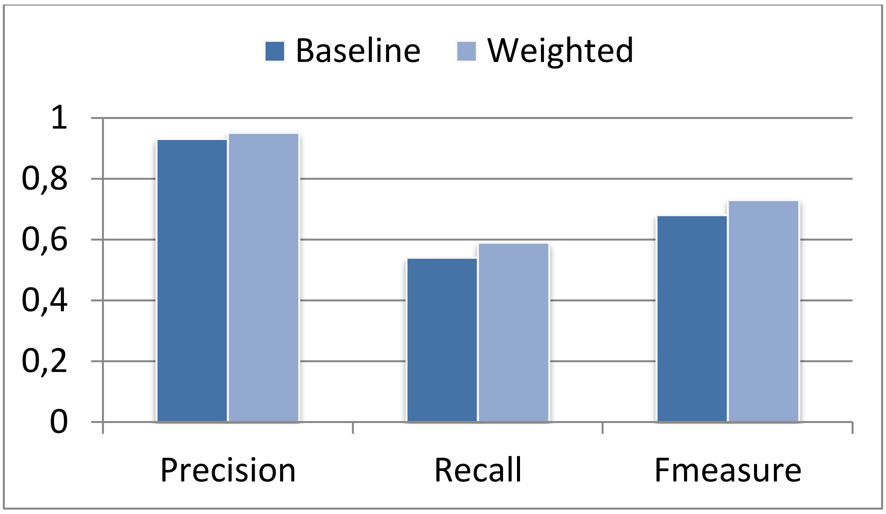


Figure 13: Baseline vs proposed weighted approach of ontology alignment against the OAEI bibliographic benchmark dataset

According to the indicators defined in D1.2 where the lowest and highest expectations for Task 4.2 are listed, the improvement that is expected over the baseline versions is between 3%-5% in F-measure. The results indicate an improvement of 5% so it is within the expected improvement values.

5.3 Source code availability

Complying with the requirements of MULTISENSOR to publish the source code of the developed tools online, the extensions of the Alignment API and the user interface (D4.2) that has been developed can be found at:

https://github.com/MKLab-ITI/multisensor-ontology-alignment

6 CONCLUSIONS

D4.4 reported on the progress of Task 4.2 and Task 4.3 on WP4. Regarding Task 4.2, the developments that were related to the extension of the Alignment API for efficiently supporting the weighted fusion approach is described. The source code has been made publicly available on GitHub[9]. Regarding Task 4.3, the development of an entity alignment service, as part of the CEP, is described where it is used for identifying and removing contradictory entities. In addition, the final version of the CAP was described along with all the defined similarity and contradiction measures. An evaluation has been carried out and documented on a MULTISENSOR dataset. The EA module and CAP are language agnostic as they rely on information that has been extracted from article processing modules in the CEP. As they don't access and work on the raw text, they can be adapted to any language (the modules are language independent).

[9] GitHub, http://www.github.com

7 REFERENCES

Alexiev, V. and Casamayor, G., 2016, May. FN goes NIF: Integrating FrameNet in the NLP Interchange Format. In LDL 2016 5th Workshop on Linked Data in Linguistics: Managing, Building and Using Linked Language Resources (p. 1).

Baker, C.F., Fillmore, C.J. and Lowe, J.B., 1998, August. The berkeley framenet project. In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1 (pp. 86-90). Association for Computational Linguistics.

Bizer, C., Heath, T. and Berners-Lee, T., 2009. Linked data-the story so far. Semantic Services, Interoperability and Web Applications: Emerging Concepts, pp.205-227.

Euzenat, J., 2004, November. An API for ontology alignment. In International Semantic Web Conference (pp. 698-712). Springer Berlin Heidelberg.

Fillmore, C., 1982. Frame semantics. Linguistics in the morning calm, pp.111-137.

Ji, H. and Grishman, R., 2011, June. Knowledge base population: Successful approaches and challenges. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (pp. 1148-1158). Association for Computational Linguistics.

Knuth, M., Hercher, J. and Sack, H., 2012. Collaboratively patching linked data. arXiv preprint arXiv:1204.2715.

Nikolić, M., 2012. Measuring similarity of graph nodes by neighbor matching. Intelligent Data Analysis, 16(6), pp.865-878.

Paulheim, H. and Bizer, C., 2013, October. Type inference on noisy rdf data. In International Semantic Web Conference (pp. 510-525). Springer Berlin Heidelberg.

Paulheim, H. and Bizer, C., 2014. Improving the quality of linked data using statistical distributions. International Journal on Semantic Web and Information Systems (IJSWIS), 10(2), pp.63-86.

Petruck, M.R., 1996. Frame semantics. Handbook of pragmatics, pp.1-13.

Ruppenhofer, J., Ellsworth, M., Petruck, M.R., Johnson, C.R. and Scheffczyk, J., 2006. FrameNet II: Extended theory and practice.