This report provides a formal description of each community driven application that will be produced by the OpenMinTeD project. We provide formal designs for 14 applications from the 4 thematic areas of the project.
D9.2 – Deliverable Name as it appears in the DOA

WP9 – Community Driven Applications Implementation (26)

WP participating organizations: ARC, University of Manchester, UKP-TUDA, INRA, EMBL, Agro-Know I.K.E, OU, EPFL, CNIO, USFD, GESIS, GRNET, Frontiers

Contractual Delivery Date: 07/2017  
Actual Delivery Date: 09/2017

Nature: Report  
Version: 1.0(Public)

Public Deliverable

Preparation slip

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Matt Shardlow</td>
<td>UNIMAN</td>
<td>31/07/2017</td>
</tr>
<tr>
<td>Robert Bossy</td>
<td>INRA</td>
<td></td>
</tr>
<tr>
<td>Mouhamadou Ba</td>
<td>INRA</td>
<td></td>
</tr>
<tr>
<td>Estelle Chaix</td>
<td>INRA</td>
<td></td>
</tr>
<tr>
<td>Sophie Aubin</td>
<td>INRA</td>
<td></td>
</tr>
<tr>
<td>Louise Deleger</td>
<td>INRA</td>
<td></td>
</tr>
<tr>
<td>Jean-Baptiste Bohuon</td>
<td>INRA</td>
<td></td>
</tr>
<tr>
<td>Claire Nédellec</td>
<td>INRA</td>
<td></td>
</tr>
<tr>
<td>Panagiotis Zervas</td>
<td>AK</td>
<td></td>
</tr>
<tr>
<td>Pythagoras Karampiperis</td>
<td>AK</td>
<td></td>
</tr>
<tr>
<td>Masoud Kiaeeha</td>
<td>UKP-TUDA</td>
<td></td>
</tr>
<tr>
<td>Andrea Zielinski</td>
<td>GESIS</td>
<td></td>
</tr>
<tr>
<td>Mappet Walker</td>
<td>Frontiers</td>
<td></td>
</tr>
<tr>
<td>Jonas Berdoz</td>
<td>Frontiers</td>
<td></td>
</tr>
<tr>
<td>Renaud Richardet</td>
<td>Frontiers</td>
<td></td>
</tr>
<tr>
<td>Lucas Ananastiou</td>
<td>OU</td>
<td></td>
</tr>
<tr>
<td>Ioannis Foufoulas</td>
<td>ARC</td>
<td></td>
</tr>
</tbody>
</table>
### Document change record

<table>
<thead>
<tr>
<th>Issue</th>
<th>Item</th>
<th>Reason for Change</th>
<th>Author</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0.1</td>
<td>Draft version</td>
<td>First Draft</td>
<td>MS</td>
<td>UNIMAN</td>
</tr>
<tr>
<td>V0.2</td>
<td>Updated Draft</td>
<td>Material incorporated from partners. Checked and regularised</td>
<td>MS</td>
<td>UNIMAN</td>
</tr>
<tr>
<td>V0.3</td>
<td>Updated Draft</td>
<td>Incorporated comments from PP, fixed formatting, etc.</td>
<td>MS</td>
<td>UNIMAN</td>
</tr>
<tr>
<td>V0.4</td>
<td>Updated Draft</td>
<td>Formatting, etc.</td>
<td>PP</td>
<td>UNIMAN</td>
</tr>
<tr>
<td>V0.5</td>
<td>Post-Review Draft</td>
<td>Incorporated Reviewer Feedback</td>
<td>MS PP RB MB EC SA LD CN PZ PK MK MN</td>
<td>UNIMAN, INRA, AK, UKP-TUDA, GESIS</td>
</tr>
<tr>
<td>V1.0</td>
<td>First version</td>
<td></td>
<td>MS PP</td>
<td>UNIMAN</td>
</tr>
</tbody>
</table>

#### D9.2

<table>
<thead>
<tr>
<th>Issue</th>
<th>Item</th>
<th>Reason for Change</th>
<th>Author</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0.1</td>
<td>Draft Version</td>
<td>Incorporated Material from D9.1</td>
<td>MS</td>
<td>UNIMAN</td>
</tr>
<tr>
<td>V0.2</td>
<td>Draft Version</td>
<td>Incorporated Updated Chapters for, LS-B and SS-A. Updated introduction</td>
<td>MS</td>
<td>UNIMAN</td>
</tr>
<tr>
<td>V0.3</td>
<td>Draft Version</td>
<td>Incorporated updated chapters for AS-C, AS-D, AS-E, LS-C, SC-B, SC-D and SC-E.</td>
<td>MS</td>
<td>UNIMAN</td>
</tr>
<tr>
<td>V0.4</td>
<td>Draft Version</td>
<td>Incorporated minor changes from INRA, UKP TUDA, EPFL and AK</td>
<td>CN MK CO PZ</td>
<td>INRA UKP-TUDA EPFL AK</td>
</tr>
<tr>
<td>Version</td>
<td>Status</td>
<td>Description</td>
<td>MS</td>
<td>UNIMAN</td>
</tr>
<tr>
<td>---------</td>
<td>----------------</td>
<td>--------------------------------------</td>
<td>----</td>
<td>--------</td>
</tr>
<tr>
<td>V0.5</td>
<td>Review Version</td>
<td>Prepared deliverable for review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V0.6</td>
<td>Post-Review</td>
<td>Incorporated Reviewer Feedback</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V1.0</td>
<td>First Version</td>
<td>For Delivery</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table of Contents

1. INTRODUCTION ...................................................................................................................... 13
   1.1 BACKGROUND ...................................................................................................................... 13
   1.2 METHODOLOGY .................................................................................................................. 13
   1.3 USE CASES .......................................................................................................................... 15
   1.4 INTEGRATION SCHEDULE ................................................................................................. 16
       1.4.1 FIRST APPLICATION RELEASE (MS37, JANUARY 2017): ........................................... 17
       1.4.2 SECOND APPLICATION RELEASE (MS49, AUGUST 2017, POSTPONED): ......................... 17
       1.4.3 THIRD APPLICATION RELEASE (MS57, FEBRUARY 2018): ......................................... 17

2. LS-A: EXTRACT METABOLITES AND THEIR PROPERTIES AND MODES OF ACTIONS .......... 18
   2.1 RESOURCES ......................................................................................................................... 19
       2.1.1 DOCUMENT FORMATS .................................................................................................. 19
       2.1.2 KNOWLEDGE BASES ................................................................................................... 19
       2.1.3 TOOLS/COMPONENTS .................................................................................................. 20
       2.1.4 SERVICES ..................................................................................................................... 21
   2.2 DEPLOYMENT PLAN ............................................................................................................. 21
   2.3 DATA INTERFACES ............................................................................................................. 22
   2.4 USER INTERFACES ............................................................................................................. 22
   2.5 DATA PROCESSING SCENARIOS ......................................................................................... 22
       2.5.1 ADDING DOCUMENTS TO THE DATABASE .................................................................. 22
       2.5.2 SHOWING ANNOTATIONS OF A GIVEN DOCUMENT .................................................... 23
       2.5.3 RETRIEVING RELATIONS FROM THE DATABASE ....................................................... 23
   2.6 LIMITATIONS ..................................................................................................................... 23
   2.7 RELEASE PLAN ................................................................................................................... 23

3. LS-B – TEXT MINING FOR CURATION OF NEUROSCIENCE LITERATURE ................................. 25
   3.1 RESOURCES ....................................................................................................................... 26
       3.1.1 DOCUMENT FORMATS .................................................................................................. 26
       3.1.2 KNOWLEDGE BASES ................................................................................................... 26
       3.1.3 TOOLS/COMPONENTS .................................................................................................. 26
       3.1.4 SERVICES ..................................................................................................................... 27
   3.2 DEPLOYMENT PLAN ............................................................................................................ 27
   3.3 DATA INTERFACES ............................................................................................................. 28
   3.4 USER INTERFACES ............................................................................................................. 28
   3.5 DATA PROCESSING SCENARIOS ......................................................................................... 28
       3.5.1 ADDING DOCUMENTS TO THE DATABASE .................................................................. 28
       3.5.2 SHOWING ANNOTATIONS OF A GIVEN DOCUMENT .................................................... 29
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.6</td>
<td>LIMITATIONS</td>
<td>29</td>
</tr>
<tr>
<td>3.7</td>
<td>RELEASE PLAN</td>
<td>29</td>
</tr>
<tr>
<td>3.8</td>
<td>REFERENCES</td>
<td>30</td>
</tr>
<tr>
<td>4.</td>
<td>LS-C: TEXT MINING ON ARTICLES RELATED TO HEALTH STATE MODELLING IN CHRONIC LIVER DISEASES</td>
<td>31</td>
</tr>
<tr>
<td>4.1</td>
<td>RESOURCES</td>
<td>33</td>
</tr>
<tr>
<td>4.2</td>
<td>DEPLOYMENT PLAN</td>
<td>33</td>
</tr>
<tr>
<td>4.3</td>
<td>DATA INTERFACES</td>
<td>33</td>
</tr>
<tr>
<td>4.4</td>
<td>USER INTERFACES</td>
<td>34</td>
</tr>
<tr>
<td>4.5</td>
<td>DATA PROCESSING SCENARIANS</td>
<td>34</td>
</tr>
<tr>
<td>4.6</td>
<td>LIMITATIONS</td>
<td>34</td>
</tr>
<tr>
<td>4.7</td>
<td>RELEASE PLAN</td>
<td>34</td>
</tr>
<tr>
<td>5.</td>
<td>AS-A: TEXT MINING OVER BIBLIOGRAPHIC DATA: AGRIS &amp; CORE</td>
<td>36</td>
</tr>
<tr>
<td>5.1</td>
<td>RESOURCES</td>
<td>36</td>
</tr>
<tr>
<td>5.1.1</td>
<td>DOCUMENT FORMATS</td>
<td>36</td>
</tr>
<tr>
<td>5.1.2</td>
<td>KNOWLEDGE BASES</td>
<td>37</td>
</tr>
<tr>
<td>5.1.3</td>
<td>TOOLS/COMPONENTS</td>
<td>37</td>
</tr>
<tr>
<td>5.1.4</td>
<td>SERVICES</td>
<td>38</td>
</tr>
<tr>
<td>5.2</td>
<td>DEPLOYMENT PLAN</td>
<td>39</td>
</tr>
<tr>
<td>5.3</td>
<td>DATA INTERFACES</td>
<td>39</td>
</tr>
<tr>
<td>5.4</td>
<td>USER INTERFACES</td>
<td>39</td>
</tr>
<tr>
<td>5.5</td>
<td>DATA PROCESSING SCENARIANS</td>
<td>40</td>
</tr>
<tr>
<td>5.6</td>
<td>LIMITATIONS</td>
<td>40</td>
</tr>
<tr>
<td>5.7</td>
<td>RELEASE PLAN</td>
<td>41</td>
</tr>
<tr>
<td>6.</td>
<td>AS-B: TEXT-MINING OVER RSS FEEDS: FOOD SAFETY &amp; WATER HEALTH</td>
<td>42</td>
</tr>
<tr>
<td>6.1</td>
<td>RESOURCES</td>
<td>42</td>
</tr>
<tr>
<td>6.1.1</td>
<td>DOCUMENT FORMATS</td>
<td>42</td>
</tr>
<tr>
<td>6.1.2</td>
<td>KNOWLEDGE BASES</td>
<td>42</td>
</tr>
<tr>
<td>6.1.3</td>
<td>TOOLS/COMPONENTS</td>
<td>43</td>
</tr>
<tr>
<td>6.1.4</td>
<td>SERVICES</td>
<td>44</td>
</tr>
<tr>
<td>6.2</td>
<td>DEPLOYMENT PLAN</td>
<td>44</td>
</tr>
<tr>
<td>6.3</td>
<td>DATA INTERFACES</td>
<td>45</td>
</tr>
<tr>
<td>6.4</td>
<td>USER INTERFACES</td>
<td>45</td>
</tr>
<tr>
<td>6.5</td>
<td>DATA PROCESSING SCENARIANS</td>
<td>45</td>
</tr>
<tr>
<td>6.6</td>
<td>LIMITATIONS</td>
<td>46</td>
</tr>
<tr>
<td>6.7</td>
<td>RELEASE PLAN</td>
<td>46</td>
</tr>
<tr>
<td>7.</td>
<td>AS-C: MICROBIAL BIODIVERSITY</td>
<td>48</td>
</tr>
<tr>
<td>7.1</td>
<td>RESOURCES</td>
<td>49</td>
</tr>
</tbody>
</table>
9.5 DATA PROCESSING SCENARIOS ........................................................................................................... 110
9.6 LIMITATIONS .................................................................................................................................. 113
9.7 RELEASE PLAN ................................................................................................................................. 113
9.7.1 1ST RELEASE (JANUARY 2017) .................................................................................................. 113
9.7.2 2ND RELEASE (AUGUST 2017) ................................................................................................. 114
9.7.3 3RD RELEASE (JANUARY 2018) ................................................................................................. 114
9.8 REFERENCES ..................................................................................................................................... 115

10. SS-A: FACILITATION OF COMPLEX INFORMATION LINKING AND RETRIEVAL FROM SOCIAL SCIENCES
PUBLICATIONS ......................................................................................................................................... 117
10.1 RESOURCES ...................................................................................................................................... 117
10.1.1 DOCUMENT FORMATS ........................................................................................................... 117
10.1.2 KNOWLEDGE BASES .............................................................................................................. 118
INTERNAL RESOURCES .......................................................................................................................... 118
EXTERNAL RESOURCES .......................................................................................................................... 120
10.1.3 TOOLS/COMPONENTS ........................................................................................................... 120
10.1.4 SERVICES .................................................................................................................................. 125
10.1.5 OTHER ....................................................................................................................................... 126
10.2 DEPLOYMENT PLAN ......................................................................................................................... 126
10.3 DATA INTERFACES ............................................................................................................................ 126
10.4 USER INTERFACES ........................................................................................................................... 127
10.5 DATA PROCESSING SCENARIOS ................................................................................................. 128
10.6 LIMITATIONS .................................................................................................................................. 130
10.7 RELEASE PLAN ............................................................................................................................... 131

11. SC-A: RESEARCH ANALYTICS – FUNDING MINING SERVICES ....................................................... 133
11.1 RESOURCES ..................................................................................................................................... 133
11.1.1 DOCUMENT FORMATS ........................................................................................................... 133
11.1.2 KNOWLEDGE BASES .............................................................................................................. 133
11.1.3 TOOLS/COMPONENTS ........................................................................................................... 133
11.1.4 SERVICES .................................................................................................................................. 134
11.2 DEPLOYMENT PLAN ......................................................................................................................... 134
11.3 DATA INTERFACES ............................................................................................................................ 134
11.4 USER INTERFACES ........................................................................................................................... 134
11.5 DATA PROCESSING SCENARIOS ................................................................................................. 135
11.5.1 APPLICATION UPDATES .......................................................................................................... 135
11.5.2 RESOURCE UPDATES ............................................................................................................ 135
11.6 LIMITATION ..................................................................................................................................... 135
11.7 RELEASE PLAN ............................................................................................................................... 135
11.7.1 JANUARY 2017 ......................................................................................................................... 135
12. SC-B: RESEARCH PUBLICATIONS RECOMMENDATION SYSTEM .......................................................... 136

12.1.1 DOCUMENT FORMATS ................................................................................................................. 136
12.1.2 KNOWLEDGE BASES ..................................................................................................................... 136
12.1.3 TOOLS/COMPONENTS .................................................................................................................... 138
12.1.4 SERVICES ........................................................................................................................................ 139
12.2 DEPLOYMENT PLAN ............................................................................................................................ 140
12.3 DATA INTERFACES ............................................................................................................................. 140
12.4 USER INTERFACES ............................................................................................................................ 140
12.5 DATA PROCESSING SCENARIOS ......................................................................................................... 141
12.6 LIMITATIONS ...................................................................................................................................... 142
12.7 RELEASE PLAN .................................................................................................................................. 143

13. SC-C: RESEARCH PERFORMANCE ...................................................................................................... 144

13.1.1 DOCUMENT FORMATS ................................................................................................................. 144
13.1.2 KNOWLEDGE BASES ..................................................................................................................... 144
13.1.3 TOOLS/COMPONENTS .................................................................................................................... 147
13.2 DEPLOYMENT PLAN ............................................................................................................................ 148
13.3 DATA INTERFACES ............................................................................................................................. 148
13.4 USER INTERFACES ............................................................................................................................ 148
13.5 DATA PROCESSING SCENARIOS ......................................................................................................... 148
13.6 LIMITATIONS ...................................................................................................................................... 149
13.7 RELEASE PLAN .................................................................................................................................. 149

14. SC-D TEXT MINING OF ARTICLES RELATED TO LEICA MICROSCOPES ............................................. 150

14.1 RESOURCES ........................................................................................................................................ 151
14.2 DEPLOYMENT PLAN ............................................................................................................................ 151
14.3 DATA INTERFACES ............................................................................................................................. 151
14.4 USER INTERFACES ............................................................................................................................ 151
14.5 DATA PROCESSING SCENARIOS ......................................................................................................... 151
14.6 LIMITATIONS ...................................................................................................................................... 152
14.7 RELEASE PLAN .................................................................................................................................. 152

15. SC-E: ROCK ART MINING ...................................................................................................................... 153

15.1 RESOURCES ........................................................................................................................................ 153
15.1.1 DOCUMENT FORMATS ................................................................................................................. 153
15.1.2 KNOWLEDGE BASES ..................................................................................................................... 154
15.1.3 TOOLS/COMPONENTS .................................................................................................................... 154
15.1.4 SERVICES ........................................................................................................................................ 155
15.1.5 OTHER ............................................................................................................................................. 155
### Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.2</td>
<td>DEPLOYMENT PLAN</td>
<td>155</td>
</tr>
<tr>
<td>15.3</td>
<td>DATA INTERFACES</td>
<td>155</td>
</tr>
<tr>
<td>15.4</td>
<td>USER INTERFACES</td>
<td>156</td>
</tr>
<tr>
<td>15.5</td>
<td>DATA PROCESSING SCENARIOS</td>
<td>156</td>
</tr>
<tr>
<td>15.6</td>
<td>LIMITATIONS</td>
<td>157</td>
</tr>
<tr>
<td>15.7</td>
<td>RELEASE PLAN</td>
<td>158</td>
</tr>
<tr>
<td>16.</td>
<td>AS-C APPENDICES</td>
<td>159</td>
</tr>
<tr>
<td>17.</td>
<td>APPENDICES OF AS-D</td>
<td>182</td>
</tr>
<tr>
<td>18.</td>
<td>APPENDICES OF AS-E</td>
<td>203</td>
</tr>
</tbody>
</table>
Disclaimer

This document contains description of the OpenMinTeD project findings, work and products. Certain parts of it might be under partner Intellectual Property Right (IPR) rules so, prior to using its content please contact the consortium head for approval.

In case you believe that this document harms in any way IPR held by you as a person or as a representative of an entity, please do notify us immediately.

The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any sort of responsibility that might occur as a result of using its content.

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of the OpenMinTeD consortium and can in no way be taken to reflect the views of the European Union.

The European Union is established in accordance with the Treaty on European Union (Maastricht). There are currently 28 Member States of the Union. It is based on the European Communities and the member states cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice and the Court of Auditors. (http://europa.eu/)

OpenMinTeD is a project funded by the European Union (Grant Agreement No 654021).
# Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDM</td>
<td>Text and Data Mining</td>
</tr>
<tr>
<td>LS</td>
<td>Life Sciences</td>
</tr>
<tr>
<td>SC</td>
<td>Scholarly Communication</td>
</tr>
<tr>
<td>AS</td>
<td>Agriculture / Biodiversity</td>
</tr>
<tr>
<td>SS</td>
<td>Social Sciences</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>POS</td>
<td>Part of Speech</td>
</tr>
<tr>
<td>IE</td>
<td>Information Extraction</td>
</tr>
<tr>
<td>IR</td>
<td>Information Retrieval</td>
</tr>
<tr>
<td>NER</td>
<td>Named Entity Recognition</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>OMTD</td>
<td>OpenMinTeD</td>
</tr>
<tr>
<td>KB</td>
<td>Knowledge Base</td>
</tr>
<tr>
<td>OMG</td>
<td>Object Management Group</td>
</tr>
<tr>
<td>LOD</td>
<td>Linked Open Data</td>
</tr>
<tr>
<td>GWS</td>
<td>GESIS-Wide Search</td>
</tr>
<tr>
<td>GESIS</td>
<td>German Social Science Infrastructure Services</td>
</tr>
<tr>
<td>UIMA</td>
<td>Unstructured Information Management Architecture</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>XMI</td>
<td>XML Metadata Interchange</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>SKOS</td>
<td>Simple Knowledge Organization System</td>
</tr>
<tr>
<td>ChEBI</td>
<td>Chemical Entities of Biological Interest</td>
</tr>
<tr>
<td>DCMI</td>
<td>Dublin Core Metadata Initiative</td>
</tr>
</tbody>
</table>
Publishable Summary

The OpenMinTeD project is producing a set of applications driven by the needs of the communities at the heart of our project. These applications will both serve a functional need for each community, as well as showcasing the functionality of the platform. In this report we aim to give a formal description of each application. We have asked the developers of the applications, alongside the communities who will be using the applications to detail important tools, services and other resources, their plans for the deployment of the application, the interfaces that will be necessary for the application, scenarios for processing data, risks and limitations of their approach and finally, a release plan detailing what aspects of functionality they expect to be released throughout the three releases of the project.
1. **Introduction**

In this report we describe the applications that will be built as a part of the OpenMinTeD project. The project seeks to create an open text and data-mining infrastructure with both interoperability and community engagement at its core. The applications that will be produced as a part of the project are important for community engagement. They are designed around real world problems put forward by our community partners and will showcase the functionality of the platform to other researchers in communities that can make use of the OpenMinTeD TDM infrastructure. Furthermore, the applications will be accessible through the platform to users across a broad spectrum of communities. The project’s community work is initially focusing on four key thematic areas: Scholarly Communications, Life Sciences, Agriculture / Biodiversity and Social Sciences, all of which face the infamous issue of ‘data-deluge’.

1.1 **Background**

This report is part of a series of deliverables concerning the engagement of the communities in the OpenMinTeD project. This process started with an assessment of the Communities requirements (Deliverable 4.2), which was used to influence the project’s requirements and the design of the infrastructure. Next, an initial formalisation of the use cases was completed (Milestone 20). Previously, the use cases were specified only as a few paragraphs in the project’s proposal. These use cases were further formalised and specified, through collaboration of technical and community partners (Deliverable 4.4). A feasibility study was undertaken for each use case to see what is practical with current technologies. Based upon these use cases and feasibility studies we designed a set of applications, which were presented in Deliverable 9.1. The application designs gave a concrete set of specifications for converting a use case document into an application. This deliverable (D9.2) provides an update to Deliverable 9.1, reflecting the ongoing work in the application delivery. We have added six new application designs (LS-B, LS-C, SC-B, SC-C, SC-D and SC-E). We have also updated eight of our original application designs (LS-A, AS-A, AS-B, AS-C, AS-D, AS-E and SS-A), reflecting ongoing work. One of the original application designs did not require any updating and is replicated here from D9.1 (SC-A). In total, this gives us fourteen applications that we are planning on delivering as part of the OpenMinTeD infrastructure.

1.2 **Methodology**

The task force for each thematic area of the project is made up of several institutions with experience in the given area. These partners are divided between ‘technical partners’ who provide TDM expertise and community partners who provide domain knowledge. The application design work was led by the technical partners from each thematic area, with close collaboration from the community partners. To facilitate and standardise the design of the applications, we created a chapter template, which covered a wide range of areas for the application designers to specify how they would build their application.
The sections of the template are given below, with the descriptions presented to the application designers. Each application design chapter that follows takes the structure that we explain below.

- **General Description**: The application design should be summarised in 1-2 paragraphs. If the application is going to be used as an element of a larger system (parent application), this should also be described here.

- **Resources**: This section is devoted to enumerating any resources this application may be using. Every subsection contains resources of one type, with information about their role in the application, but also restrictions limiting their use.
  - **Document Formats**: In case of formats that will be used to represent documents (both coming from repositories and including created annotations), the following information is relevant: Name, Link to specification/documentation, Intended use, If applicable: license, version, serialisation etc.
  - **Knowledge Bases**: This section should include a list of all knowledge bases used (vocabularies, ontologies, databases, lexicons etc.) with some basic information: Name, Version, Link to resource (url), Link to documentation, Intended use, Data format, Approximate size (MBs, number of entities/classes), Licence, terms of use.
  - **Tools/Components**: Here we should enumerate all programs used to process data in the application: both standalone tools and components in workflows with at least the following details: Name, Version, License, Link to documentation, Link to the code, Performed task, Input/Output formats, Deployment mode: standalone tool, web service, workflow component, VM, etc. Requirements for system environment (OS, installed libs, language interpreters), Performance requirements and limits (used memory, processing speed, multi-threading), Used static resources and models.
  - **Services**: If the application makes use of some external web services, they should be enumerated here: Name, Address, Performed Tasks, Input/Output formats, Terms of use (esp. performance limitations), Hosting institution, Type of interface (REST/SOAP/Other).
  - **Other**: This is the place to declare any external resources the application depends on that are not covered by the above categories, e.g. publication stores.

- **Deployment Plan**: This section should explain in detail, how its elements will be deployed, i.e. what components will be using what infrastructure to run. This applies to hosting resources and datasets, executing workflows, serving results to users etc. It is important to clearly distinguish elements of infrastructure provided by the OpenMinTeD platform, OpenMinTeD partners and external entities.

- **Data Interfaces**: Here we should enumerate all the ways, in which the application communicates with external world and between modules through data. For example, this includes downloading relevant publications from appropriate databases, transferring them between annotation components or uploading results to repositories. For every interface we need to know the following: Type of interface, Protocol, Format of data transferred, Level of automation.
**User Interfaces:** This section describes all the different ways in which the application will interact with users. This may include web interface for performing search, annotation tool GUI, a webpage displaying final results or download of user-readable files.

**Data Processing Scenarios:** This is the core part of the application design: it consists of a series of scenarios, each describing actions necessary to achieve a certain goal. For example it may describe a process of annotating a single document by downloading through publisher’s API, running an annotation workflow, adding references to vocabularies and finally rendering a visualisation of the result. Each scenario may contain steps of several types:

- Communication steps, where the application interacts with external data sources through data interfaces.
- Human interaction steps, when further processing depends on the results of users’ actions.
- Conditional steps, sending the data to different execution paths depending on certain conditions, e.g. data format.
- Storing steps, when the intermediate results are stored for future use.
- Publishing steps, when the final results of the processing are made available for the public for downloading or using.

This section heavily relies on the remainder of the document, as it refers to resources, communication interfaces, etc. defined elsewhere.

**Limitations:** This part should detail the limitations of the presented approach, for example insufficient speed, poor scalability, reliance on web services, licence restrictions. Furthermore, it needs be explained here what is the relation between this document and the scenarios in D4.3: whether the whole functionality is covered, what elements are excluded and why. If implementing some components covered by this design depends on currently unavailable elements of OpenMinTeD platform, it should be explained here as well.

**Release Plan:** Finally, the document should include a plan of releasing the application, detailing what functionalities are included in what versions. This should take into account the fact that the platform will be released in steps, so the first versions of application may need to rely on existing infrastructure for some elements, which will subsequently be migrated to the target environment.

### 1.3 Use Cases

The following chapters of the report each describe one application. We have fourteen application designs from the four thematic areas of the project. The titles of the applications described in this report are in the following table.
The use cases created by Frontiers (LS-C, SC-D and SC-E) are less mature than the others in the project as work on them has begun later on in the project’s lifecycle. Accordingly, the application designs for these use cases are lighter than those of other use cases. They are necessarily subject to change as Frontiers’ further concretises these applications. Finalised versions of the application design documents will be produced and incorporated into the final WP9 deliverable: D9.4 - Application Software Release Report.

1.4 Integration Schedule

The OpenMinTeD project is producing a platform, which the applications will run on. However, as the applications and the platform are being developed in parallel, the integration between the two must be staged. The level of integration at each application release is detailed below:
1.4.1 First Application Release (MS37, January 2017):
In the first application release, we did not have sufficient infrastructure to be able to deploy applications. We therefore created the applications as stand-alone services. This allowed each partner to focus on delivering the functionality of the application itself.

1.4.2 Second Application Release (MS49, August 2017, postponed):
For the second release, we plan to deliver each application running as a self-contained workflow on the OpenMinTeD infrastructure. To do this, we will put each application into a docker container. The Docker containers will follow the specification produced by WG4, ensuring that they all have a common interface. These applications will be placed in the registry to allow them to be used as part of the OpenMinTeD platform. This application release is presently postponed, pending the release of the workflow service.

1.4.3 Third Application Release (MS57, February 2018):
For the third and final application release, the applications will be released as configurable workflows. Each component in an application will be placed into a Docker image and the OpenMinTeD workflow editor will be used to configure the workflows. This will allow new users to change the functionality of a workflow to suit their needs, as well as to create new workflows from the building blocks of other workflows.
2. LS-A: Extract Metabolites and their Properties and Modes of Actions

This application will aid the curation of the ChEBI database – a manually curated resource providing information about chemical entities, which is widely used for reference purposes in bioinformatics. We will produce tools to identify specific types of named entities and relations between them. These annotations will be stored in an SQL database and a curator will be able to query the database to quickly understand the types of entities and relations present in the literature.

The application will run a named entity recogniser to detect the following entities:

- Species
- Metabolites
- Biological Activity
- Biological Target
- Spectral Data
- Chemical Structures (SMILES, InCHI, InChI Key)
- Numbering system used to identify the position of substituents
- Proteins

It will then use a relation extraction service to detect relations in the following categories:

- Isolated_from(Metabolite, Species)
- Metabolite_of(Metabolite, Species)
- Associated_with(Metabolite, Biological Activity)
- Associated_with(Metabolite, Spectral Data)
- Has_Structure(Metabolite, Chemical Structure)
- Has_Target(Metabolite, Biological Target)
- Binds_with(Metabolite, Protein)

The implementation will consist of the following modules:

- **Annotation workflow**: implemented on OpenMinTeD platform and recognising the desired entities and relations between them in an individual document.
- **Annotation viewer and store**: also using OpenMinTeD infrastructure, providing an interface for users to see documents with annotations.
- **Annotation manager**: sending user-supplied corpus to the workflow, receiving the created annotations (possibly supported by OpenMinTeD platform), extracting relevant information (relations of interest) and sending to a database.
- **Database of relations**: storing the relations between entities across all documents, including links to annotated documents with evidence.
- **Query interface**: allowing users to search for relations between entities they are interested in and returning both the information and evidence pointers.
It is worth noting that even applications consisting of only some of the above modules can still be useful for a biocurator. See the ‘Release plan’ section for more information about what elements will be included in releases.

2.1 Resources

2.1.1 Document Formats

EuropePubMedCentral XML format
- Link to specification/documentation: None available
- Intended use: Extracting contents of titles and abstracts from EuropePMC to mine for entities and relations
- Serialisation: XML

PubAnnotation format
- Link to specification/documentation: http://www.pubannotation.org/docs/annotation-format/
- Intended use: Storing entities and relations
- Serialisation: JSON

2.1.2 Knowledge Bases

NCBI Taxonomy
- Version: Latest available database dump (to be updated in D9.2)
- Intended Use: Linking (or grounding) species names to formal ontological identifiers
- Data Format: own, TSV-based
- Approx size: around 4.5GB (compressed)
- Licence: “Free to use (NCBI itself places no restrictions on the use or distribution of the data contained therein. Nor do we accept data when the submitter has requested restrictions on reuse or redistribution. However, some submitters of the original data (or the country of origin of such data) may claim patent, copyright, or other intellectual property rights in all or a portion of the data (that has been submitted). NCBI is not in a position to assess the validity of such claims and since there is no transfer of rights from submitters to NCBI, NCBI has no rights to transfer to a third party. Therefore, NCBI cannot provide comment or unrestricted permission concerning the use, copying, or distribution of the information contained in the molecular databases)” - https://www.ncbi.nlm.nih.gov/home/about/policies.shtml

Human Metabolome Database (HMDB)
- Version: 3.6
- Link to resource: http://www.hmdb.ca/downloads
- Link to documentation: http://www.hmdb.ca/metabolites/
- Intended Use: Auxiliary, for grounding recognised human metabolites
- Data Format: application specific, XML-based
- Approx size: 177 MB (compressed)
- Licence: Free to use for non-commercial purposes - http://www.hmdb.ca/about
UniProt Knowledgebase (UniProtKB)
- Version: 2016_10
- Link to resource: http://www.uniprot.org/downloads#uniprotkblink
- Link to documentation: http://www.uniprot.org/help/
- Intended Use: For grounding recognised proteins
- Data Format: own XML-based, FASTA, text
- Approx size: 59 GB (compressed)

Chemical Entities of Biological Interest (ChEBI)
- Version: Database dump dated 1st November 2016
- Link to resource: https://www.ebi.ac.uk/chebi/
- Link to documentation: https://www.ebi.ac.uk/chebi/userManualForward.do
- Intended Use: Base corpus of entities for metabolites, etc.
- Data Format: SDF / OWL / OBO / TSV / SQL (Likely to use TSV)
- Approx size: 2.4GB (TSV)
- Licence: “All data in the database is non-proprietary or is derived from a non-proprietary source. It is thus freely accessible and available to anyone. In addition, each data item is fully traceable and explicitly referenced to the original source.” - https://www.ebi.ac.uk/chebi/aboutChebiForward.do

Europe PubMed Central (Europe PMC)
- Link to resource: https://europepmc.org/
- Link to documentation: https://europepmc.org/downloads/openaccess
- Intended use: Articles for training / testing purposes
- Data format: EuropePMC XML format
- Approximate size: ~20MB
- Licence: https://europepmc.org/Copyright

NaCTeM Metabolite and Enzyme Corpus
- Version: 09/01/2013
- Link to resource: http://www.nactem.ac.uk/metabolite-corpus/
- Link to documentation: http://www.nactem.ac.uk/metabolite-corpus/
- Intended use: Grounding of metabolite mentions
- Data format: XML
- Licence: NLM licence (Pubmed), Creative Commons Attribution 3.0 Unported License (Metabolite annotations)

2.1.3 Tools/Components
Chemical Entity Recogniser (Component in Argo)
- Version: 0.1-SNAPSHOT
- License: NaCTeM Proprietary Licence (see: http://nactem.ac.uk/terms_conditions.php)
- Link to documentation: None
- Link to the code: Not available outside of NaCTeM's servers
• Performed task: recognising metabolites in biomedical text
• Input/Output formats: UIMA CAS - Argo type system
• Deployment mode: workflow component
• Requirements for system environment (OS, installed libs, language interpreters): UIMA compatible
• Performance requirements and limits (used memory, processing speed, multi-threading): Unknown
• Used static resources and models: None

**NERSuite Custom Tagger** (Component in Argo)
• Version: 0.1-SNAPSHOT
• License: NaCTeM Proprietary Licence (see: http://nactem.ac.uk/terms_conditions.php)
• Link to documentation: None
• Link to the code: Not available outside of NaCTeM’s servers
• Performed task: Customisable tagging of named entities (proteins, etc.)
• Input/Output formats: UIMA CAS - Argo type system
• Deployment mode: workflow component
• Requirements for system environment (OS, installed libs, language interpreters): UIMA compatible
• Performance requirements and limits (used memory, processing speed, multi-threading): Unknown
• Used static resources and models: Depends on application

**Species Tagger** (Component in Argo)
• Version: 0.0.1-SNAPSHOT
• License: NaCTeM Proprietary Licence (see: http://nactem.ac.uk/terms_conditions.php)
• Link to documentation: None
• Link to the code: Not available outside of NaCTeM’s servers
• Performed task: Customisable tagging of named entities (Proteins, etc.)
• Input/Output formats: UIMA CAS - Argo type system
• Deployment mode: workflow component
• Requirements for system environment: UIMA compatible
• Performance requirements and limits: Unknown
• Used static resources and models: Depends on application

2.1.4 Services
No external services are needed for this application.

2.2 Deployment Plan
As detailed in the release plan (see Section 8), the project infrastructure will follow a staged release plan. Therefore, we will not be able to implement all components of the application upon the OpenMinTeD infrastructure in the initial releases. Instead we will use the Argo platform, a UIMA based workflow manager hosted at NaCTeM. Argo provides a workflow service, similar to the one proposed in the OpenMinTeD infrastructure, which can be used to create and export annotations. That is reflected in the deployment plan below, when some of the elements have more than one location.
Components that are implemented for use in Argo will be easily transferable to the OpenMinTeD infrastructure when it becomes available for future releases.

The deployment of the application requires the following elements:

- The annotation workflow will ultimately run on OpenMinTeD platform, which will include a workflow manager. Before it becomes available, the annotation process could be ran on Argo.
- Annotation viewing and storing will be supported by OpenMinTeD infrastructure (annotation editor), but before that a limited number of documents could be viewed in NaCTeM’s installation of the Brat service, which will display results in any web browser (http://brat.nlplab.org/).
- Annotation manager - a standalone executable which can be run by the user when necessary.
- A database of relations between entities will be stored in an SQL server hosted at EBI
- Query interface will be implemented as an HTML form deployed at EBI

2.3 Data Interfaces

Workflow input/output

- Protocol: HTTP (REST WS)
- Format of data transferred: Plain text/PubAnnotation
- Level of automation: full

Database insertion and queries

- Protocol: SQL
- Format of data transferred: Tables according to schema
- Level of automation: full

2.4 User Interfaces

There are three user interfaces involved in this scenario:

1) To upload new documents for annotation the user will need to use Annotation Manager to indicate the document(s) that they wish to be annotated by uploading a single text or a whole archive.

2) Once these documents have been annotated, the user will use the OpenMinTeD annotation editor (or NaCTeM’s Brat installation) to view the annotations made by the workflow.

3) Finally, a user can query the system for all relations by specifying a relation type and/or involved entities. As a result, the system will return all relations satisfying the query and relation type and will also provide links to annotated documents with evidence. The user can then process this information for curation.

2.5 Data Processing Scenarios

2.5.1 Adding documents to the database

- A user runs Annotation Manager specifying a text document or a full PubMed archive.
• Depending on input type, the Annotation Manager initiates an appropriate workflow through HTTP interface.
• The workflow manager runs a desired workflow which finds the entities and relations in text,
• The annotation is returned in PubAnnotation format to the Annotation Manager, which stores it in Annotation viewer store.
• Annotation Manager also aggregates the discovered relations and, if necessary, amends the relation database.
• Depending on the users’ requirements, this scenario could be executed only once for a defined set of documents or periodically to keep up to date with all the literature in a field.

2.5.2 Showing annotations of a given document
• Annotation viewer displays a document selection interface to the user. The documents are ordered by PMIDs to facilitate choice,
• The user chooses one of the documents,
• Annotation viewer displays the document with annotated relations.

2.5.3 Retrieving relations from the database
• A user opens the query interface and specifies either one or both of the criteria:
  o Type of relation,
  o Participating entities (by IDs),
• Appropriate SQL query is created and sent to the database,
• The database engine finds required relations,
• The retrieved relations are displayed to the user,
• User may also click one of the evidence links, which will redirect him to the annotation viewer pre-loaded with the desired document.

2.6 Limitations
The application relies heavily on the OpenMinTeD infrastructure - hence it is critical that the specifications of the infrastructure are known well in advance to allow for the timely development of components and workflows that will work in harmony with the project’s infrastructure. The full release of the infrastructure is not set to be available until mid-way through the project, and so we will use Argo where necessary.

The application relies on the provision of annotated datasets for the requisite entities and relations. These will be provided by annotators at the EBI who are engaged with the day to day curation of the ChEBI database. The quality of the tools will depend directly on how much annotated data can be retrieved. Typical NER systems may require several thousand annotated examples to attain state of the art performance. The consistency of the annotations may also be a limiting factor. As such, we should produce clear guidelines, monitor agreement between annotators and resolve conflicts between annotators to ensure high-quality consistent annotations.

2.7 Release Plan
There are 3 key releases for the applications. These are as follows:

- Jan 2017 - 1st Application Release
- August 2017 - 2nd Application Release
- Feb 2018 - 3rd Application Release

We expect the following parts of the OpenMinTeD infrastructure to be available at each release:

- Jan 2017 - Registry Available
- August 2017 - Initial version of workflow service available
- Feb 2018 - Annotation viewing + editing service available

Consequently, the release plan for this application will be as follows:

**Jan 2017**
We will deliver the application built upon the Argo workflow management system. The components that we use will be registered in the OpenMinTeD registry. The annotations will focus primarily on the named entities which are of interest to us in the annotation of ChEBI. Annotations will be displayed within Argo.

**August 2017**
We will deliver the application running upon the workflow service provided by OpenMinTeD. The application will be adapted to implement relation extraction between the entity types of interest. The Annotations will be displayed in a third party tool such as Brat or WebAnno.

**Feb 2018**
The database will be populated with the annotations produced by the workflow. A search functionality will be created over this database which will allow curators to view the types of information that they are interested in for different metabolites.
3. **LS-B – Text Mining for Curation of Neuroscience Literature**

This application is focused on supporting the curation of neuroscience entities from the literature with the aim of supporting ongoing curation efforts in the Blue Brain Project (BBP), at the École Polytechnique Fédérale de Lausanne (EPFL). For that purpose, a manual curation framework has been designed (O’Reilly 2017). This framework allows to create corpora of curated literature facts and of parameter values published in scholarly journals. These corpora are then queried to fill in numerical values to parameters used in large scale biophysically-detailed models of the brain. Importantly, annotations produced in this context provide full traceability (down to the specific words of a specific documents) of the information embedded in the modeling process. It also offers a collaborative framework for curation tasks and promote the reuse of literature annotations by linking every annotated fact or parameter to ontology entities defined in the Neuroscience Information Framework (NIF) ontology. This ontology is publicly available via the web (https://neuinfo.org/) and provides information about a wide number of neuroscience entities. However, one limitation to this framework is that the curation need be performed manually and this process is labor-intensive. Thus, in context of this use case, we aim to produce tools to recognize important entities in neuroscience to help automatize the curation process. These annotations will be stored in corpora which can thereafter be queried to extract the best values from literature to model different biological systems in neuroscience. With a critical mass of annotations, they also can start to be used to answer important questions such as the impact of various factors (species, experimental procedure, age, etc.) on some parameter values (e.g., densities of neuron types in the thalamus).

The application will provide named entity recognition for the following entity types:

- Neurons
- Brain Regions
- Synapses
- Species / Model Organisms
- Experimental Values + Units
- Ionic Channels / Conductances / Currents

The implementation will consist of the following modules:

- **Annotation workflow**: implemented on OpenMinTeD platform and recognising the desired entities and relations between them in an individual document.
- **Annotation viewer and store**: also using OpenMinTeD infrastructure, providing an interface for users to see documents with annotations.
- **Annotation manager**: sending user-supplied corpus to the workflow, receiving the created annotations (possibly supported by OpenMinTeD platform), extracting relevant information (relations of interest) and sending to a database.
D9.2 – Community Driven Applications Design Report

- **Database of relations**: storing the relations between entities across all documents, including links to annotated documents with evidence.
- **Query interface**: allowing users to search for relations between entities they are interested in and returning both the information and evidence pointers.

## 3.1 Resources

### 3.1.1 Document Formats

We will support the ingestion of documents in the following formats

We will create corpora based on both abstracts and full texts to ensure robust classification accuracy on out test documents.

**EuropePubMedCentral XML format**

- Link to specification/documentation: None available
- Intended use: Extracting contents of titles and abstracts from EuropePMC to mine for entities and relations
- Serialisation: XML

**PubAnnotation format**

- Intended use: Storing entities and relations
- Serialisation: JSON

### 3.1.2 Knowledge Bases

**NIF Ontology**

- Link to documentation: [https://neuinfo.org/about/nifvocabularies](https://neuinfo.org/about/nifvocabularies)
- Intended Use: Lists of names for entity types to enable dictionary matching
- Data Format: ttl (turtle), also linked to external ontologies in OWL format
- Approx Size: Unknown
- Licence: CC-BY

### 3.1.3 Tools/Components

**NERSuite Dictionary Matcher** (Component in Argo)

- Version: 0.1-SNAPSHOT
3.2 Deployment Plan

The application will make use of the OpenMinTeD infrastructure as much as possible. We envision using the OpenMinTeD workflow service and annotation viewer to run and visualise the annotations.

The deployment of the application requires the following elements:
The annotation workflow will ultimately run on OpenMinTeD platform, which will include a workflow manager. Before it becomes available, the annotation process could be run using Argo.

Annotation viewing and storing will be supported by OpenMinTeD infrastructure (annotation editor), but before that a limited number of documents could be viewed in NaCTeM’s installation of the Brat service, which will display results in any web browser (http://brat.nlplab.org/).

Annotation manager - a standalone executable, which can be run by the user when necessary.

A database of relations between entities.

3.3 Data Interfaces

Workflow input/output
- Protocol: HTTP (REST WS)
- Format of data transferred: Plain text/PubAnnotation
- Level of automation: full

Database insertion and queries
- Protocol: SQL
- Format of data transferred: Tables according to schema
- Level of automation: full

3.4 User Interfaces

There are three user interfaces involved in this scenario:

1) To upload new documents for annotation the user will need to use Annotation Manager to indicate the document(s) that they wish to be annotated by uploading a single text or a whole archive.

2) Once these documents have been annotated, the user will use the OpenMinTeD annotation editor (or NaCTeM’s Brat installation) to view the annotations made by the workflow.

3) Finally, a user can query the system for all relations by specifying a relation type and/or involved entities. As a result, the system will return all relations satisfying the query and relation type and will also provide links to annotated documents with evidence. The user can then process this information for curation.

3.5 Data Processing Scenarios

3.5.1 Adding documents to the database
- A user runs Annotation Manager specifying a text document or a full PubMed archive.
- Depending on input type, the Annotation Manager initiates an appropriate workflow through HTTP interface.
- The workflow manager runs a desired workflow which finds the entities and relations in text,
- The annotation is returned in PubAnnotation format to the Annotation Manager, which stores it in Annotation viewer store.
3.5.2 Showing annotations of a given document

- Annotation viewer displays a document selection interface to the user. The documents are ordered by PMIDs to facilitate choice,
- The user chooses one of the documents,
- Annotation viewer displays the document with annotated relations.

3.6 Limitations

The application relies heavily on the OpenMinTeD infrastructure - hence it is critical that the specifications of the infrastructure are known well in advance to allow for the timely development of components and workflows that will work in harmony with the project’s infrastructure. The full release of the infrastructure is not set to be available until mid-way through the project, and so we will use Argo where necessary.

The application relies on the provision of annotated datasets for the requisite entities and relations. These will be provided by annotators at EPFL. The quality of the tools will depend directly on how much annotated data can be retrieved. Typical NER systems may require several thousand annotated examples to attain state of the art performance. The consistency of the annotations may also be a limiting factor. As such, we should produce clear guidelines, monitor agreement between annotators and resolve conflicts between annotators to ensure high-quality consistent annotations.

Much of the information that would be most useful for text mining in the neuroscience domain is locked away in articles that are distributed as either scans of printed version of papers, or are found behind a paywall. Either way, these documents are presently unavailable on the OpenMinTeD platform. This limitation can be overcome by hosting the documents locally and running the tools on them outside of the OpenMinTeD platform. However, issues may still exist with the quality of the OCR and the legal issues surrounding the copyright of closed access documents.

3.7 Release Plan

There are 2 key releases for this application. These are as follows:

- August 2017 - 2nd Applications Release
- Feb 2018 - 3rd Applications Release

We expect the following parts of the OpenMinTeD infrastructure to be available at each release:
Consequently, the release plan for this application will be as follows:

**August 2017**

We will deliver the application running upon the workflow service provided by OpenMinTeD. The application will provide entity recognition for the entity types of interest. The Annotations will be displayed in a third party tool such as Brat or WebAnno.

**Feb 2018**

The database will be populated with the annotations produced by the workflow. A search functionality will be created over this database which will allow curators to view the types of information that they are interested in for different metabolites. The workflow will be deployed as a configurable workflow on the OpenMinTeD infrastructure.

### 3.8 References

4. LS-C: Text mining on articles related to Health State Modelling in Chronic Liver Diseases

Health State Modelling represents different health states to provide the right intervention for the right patient at the right time and dose.

Modeling of such health states should allow iterative optimization, as human data accumulate. The platform is intended to facilitate the discovery of links between opportunities related to a) the modernization of diagnosis, b) patient-centric approaches enabled by technology convergence, including digital health and connected devices, c) increasing understanding of the pathobiological, clinical and health economic aspects of disease progression stages, d) design of new interventions, including therapies as well as preventive measures, including sequential intervention approaches.

Figure 1, below, shows where health states fit into the wider ecosystem of chronic diseases, and how health states may be used to link multiple stakeholders.

This application design document represents Frontiers’ efforts so far and is subject to further updates according to concretization of the application’s requirements.

*Figure 1* The proposed innovation ecosystem for chronic diseases, with a new platform that engages different health innovation stakeholders, and allows the emergence of interdisciplinary understanding of health states across biology, medicine, and health economics
In this system, Text and Data Mining is a tool to aid the definition of Chronic Diseases Health States. As a prototype and proof-of-concept, OpenMinTED extracts Chronic Liver Diseases articles from the data repository to identify article based health states and interventions. The first prototype focuses on Non-Alcoholic Fatty Liver Diseases and subsequent health states, such as Non-Alcoholic Steatohepatitis and End-state liver disease.

Figure 2, below, shows the role of text and data mining in modeling health states for chronic liver disease. Each of the elements in the diagram (diseases, interventions, outcomes), can be extracted automatically from the literature.

Programmatically, the TDM steps are as follows (shown diagrammatically in Figure 3):

Step #1: TDM program extracts NAFLD - Non-Alcoholic Fatty Liver Disease articles from OMTD the data repository

Step #2: TDM program downloads the extracted NAFLD articles

Step #3: TDM program annotates the articles based on keywords.

Step #4: TDM program suggests connectors and transition terms.
Step #6 – As the user confirms relevant articles, connectors and transition terms, the dictionary of features and connectors is updated.

Step 1-6 is repeated with the updated dictionary as input to the subsequent extraction activities.

4.1 Resources
The HSM Chronic Liver Disease proof of concept is at the very early stages of requirements and design. Using agile methodology, resources (e.g. document formats, knowledge bases, tools/components, services, and other resources) will be defined as the first proof-of-concept is developed in September.

4.2 Deployment Plan

The first proof-of concept will be a stand-alone program running independently of OMTD using some of its available tools. As the OMTD platform and interfaces become available, the HSM Chronic Liver Disease programs will be executed from the platform with the specified APIs and components.

This section will be updated as the OMTD platform and interfaces become available.

4.3 Data Interfaces
As of this writing, the main interface is to the CORE data repository to extract Chronic Liver Disease related articles, to develop a dictionary of features and connectors and suggest a Health State Matrix.

The details (e.g. type of interface, protocol, format of data transferred, and level of automation) will be updated as the proof-of-concept matures into a full-blown HSM Chronic Liver Disease TDM application.

4.4 User Interfaces

The details of the User Interface will be designed and developed after the main extract, annotate, define dictionary of features, connectors, and transition terms are available sometime December 2017.

4.5 Data Processing Scenarios

The Data Processing Scenarios results from the iterations on the prototypes and will be defined with the users and SMEs.

The first pass scenario reads the data repository of articles, extracts articles with Non-Alcoholic Fatty Liver Diseases, identifies potential connectors and the state transition. This is stored on a file for user confirmation. The process iterates until the number of articles reaches a level where a health state matrix can be derived.

The second and subsequent scenario is defined by the results of the first pass scenario.

4.6 Limitations

Text and Data Mining Assisted Health State Modelling is in its infancy. The results are limited by available data to mine, available SMEs to liaise with and the functionality available from the interface to the OMTD platform.

This prototype handles Non-Alcoholic Fatty Liver Diseases as a form of Chronic Liver Disease only based on articles available in the CORE repository. It does not handle other forms of Chronic Liver Diseases such as Autoimmune Hepatitis, Alcohol related liver-disease, Primary sclerosing cholangitis, primary biliary cirrhosis, to name a few.

It will be implemented as a stand-alone Text and Data Mining enable Chronic Liver Disease prototype and will be integrated into OMTD as the APIs and components become available.

4.7 Release Plan
A first version of the prototype will be available by early September based on the first pass scenario. This will be updated on a regular basis resulting from discussions with the users and SMES and availability of the OMTD platform, interfaces and components.
5. **AS-A: Text mining over bibliographic data: AGRIS & CORE**

The text-mining problem to be addressed in the context of this Use Case is the extraction of structured information (domain specific topics, images / figures, captions etc.) from unstructured bibliographic resources (i.e. PDF documents). It will provide text mining tools to identify grape varietes and terms from AgroVoc.

Two main bibliographic data sources will be used, namely, AGRIS and CORE:

- **AGRIS** (http://agris.fao.org), is an online service that provides access to more than 8 million bibliographic records for Agricultural Sciences and Technology. It covers a large number of topics of the agricultural sector. The bibliographic records are aggregated from more than 200 data providers from all around the world.

- **CORE** (https://core.ac.uk), is a service that aggregates bibliographic data from various sources including repositories, journals and archives in order to facilitate the access to open access content. It currently contains more than 37 million open access articles, from over 6000 journals, collected from over 1100 repositories around the world.

The community driven front-end application will target the viticulture community, and it will be part of VITIS (http://vitis.agroknow.com), a toolkit for supporting Viticulture practice, which ranges from web-based environments for data management, visualization and predictive model execution to mobile apps (and sensors) for in-field data collection. The front-end application will be used and tested by the Laboratory of Viticulture of the Agricultural University of Athens (AUA, http://www.aua.gr). Laboratory of Viticulture has been serving the viticulture science producing high quality research in topics such as clonal selection, classification of grape cultivars, identification and genetic study of Greek grape cultivars.

### 5.1 Resources

#### 5.1.1 Document Formats

<table>
<thead>
<tr>
<th>Name</th>
<th>Specification</th>
<th>Intended Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>JavaScript Object Notation (JSON)</td>
<td><a href="http://www.json.org/">http://www.json.org/</a></td>
<td>Text Mining</td>
</tr>
</tbody>
</table>
### 5.1.2 Knowledge Bases

<table>
<thead>
<tr>
<th></th>
<th>AGRIS</th>
<th>CORE</th>
<th>AGROVOC</th>
<th>AK-MGV</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>2.0</td>
<td>2.0</td>
<td>AGROVOC-65</td>
<td>1.0</td>
</tr>
<tr>
<td><strong>Link to resource</strong></td>
<td><a href="http://agris.fao.org">http://agris.fao.org</a></td>
<td><a href="https://core.ac.uk">https://core.ac.uk</a></td>
<td><a href="http://aims.fao.org/standards/agrovoc/concept-scheme">http://aims.fao.org/standards/agrovoc/concept-scheme</a></td>
<td>-</td>
</tr>
<tr>
<td><strong>Intended use</strong></td>
<td>Bibliographic Data Source</td>
<td>Bibliographic Data Source</td>
<td>Linguistic Resource</td>
<td>Linguistic Resource</td>
</tr>
<tr>
<td><strong>Data Format</strong></td>
<td>JSON</td>
<td>JSON</td>
<td>SKOS</td>
<td>TEXT</td>
</tr>
<tr>
<td><strong>Approx size</strong></td>
<td>26,244 approx.</td>
<td>5,470 approx.</td>
<td>20,000+ Concepts</td>
<td>4,000+ Grape Variety Terms per language (EN, FR, IT, ES, DE)</td>
</tr>
<tr>
<td><strong>Licence</strong></td>
<td>CC BY 3.0</td>
<td>-</td>
<td>CC BY 3.0</td>
<td>GPL 3.0</td>
</tr>
</tbody>
</table>

### 5.1.3 Tools/Components

<table>
<thead>
<tr>
<th></th>
<th>PDF Extractor</th>
<th>AgroVoc Extractor</th>
<th>Grape Varieties Extractor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td><strong>Licence</strong></td>
<td>GPL 3.0</td>
<td>GPL 3.0</td>
<td>GPL 3.0</td>
</tr>
<tr>
<td><strong>Link to documentation</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Link to Code</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
### Task Performed

<table>
<thead>
<tr>
<th>Extract text, figures and tables from PDF documents</th>
<th>Search and Rank Agricultural Terms in PDF documents</th>
<th>Search and Rank Grape Varieties in PDF documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDF/JSON</td>
<td>TEXT/XML</td>
<td>TEXT/XML</td>
</tr>
<tr>
<td>Standalone</td>
<td>Standalone</td>
<td>Standalone</td>
</tr>
</tbody>
</table>

### I/O Format

- PDF/JSON
- TEXT/XML

### Deployment Mode

- Standalone

### System Requirements

- a) Python v.2.7
- b) PDF Miner
  - (https://pypi.python.org/pypi/pdfminer/)
- c) Poppler
  - (https://poppler.freedesktop.org/)
- d) PDF Figures
  - (https://github.com/allenai/pdffigures/)
- a) Java Virtual Machine 1.7
- b) Lucene 6.2.1
- a) Java Virtual Machine 1.7
- b) Lucene 6.2.1

### Performance Requirements

- -
- -
- -

### Static Resources

- -
- AGROVOC-65
- AK-MGV Ontology 1.0

### 5.1.4 Services

#### Agrovoc Web Service

<table>
<thead>
<tr>
<th>Address</th>
<th>org.fao.aims.aos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Performed</td>
<td>Search the main text for AgroVoc terms</td>
</tr>
<tr>
<td>I/O Format</td>
<td>STRING/SKOS</td>
</tr>
<tr>
<td>Terms of Use</td>
<td>CC BY 3.0</td>
</tr>
<tr>
<td>Hosting Institution</td>
<td>Food and Agriculture Organization of the United Nations (FAO)</td>
</tr>
<tr>
<td>Interface Type</td>
<td>SOAP</td>
</tr>
</tbody>
</table>
5.2 Deployment Plan

The described components (PDF Extractor, AgroVoc Extractor and Grape Varieties Extractor) are deployed as standalone tools in a single virtual machine (VM). The required infrastructure is provided by the Greek Research and Technology Network (https://grnet.gr/en/) through the cloud service OKEANOS (https://okeanos.grnet.gr/home/).

The resources retrieved by the bibliographic data sources as well as the AK-MGV linguistic resource are stored in the same VM.

<table>
<thead>
<tr>
<th>Hosting Organisation</th>
<th>PDF Extractor</th>
<th>AgroVoc Extractor</th>
<th>Grape Varieties Extractor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud Service</td>
<td>Greek Research and Technology Network</td>
<td>OKEANOS</td>
<td></td>
</tr>
<tr>
<td>CPU Cores</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>RAM (GB)</td>
<td>8</td>
<td>8</td>
<td>200</td>
</tr>
<tr>
<td>Hard Disk (GB)</td>
<td>200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.3 Data Interfaces

<table>
<thead>
<tr>
<th>Interface Type</th>
<th>Bibliographic Data Ingestion</th>
<th>Publication interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>Data Stream</td>
<td>Database (MySQL)</td>
</tr>
<tr>
<td>Protocol</td>
<td>REST</td>
<td>SQL</td>
</tr>
<tr>
<td>Data transfer format</td>
<td>JSON/PDF</td>
<td>Structured</td>
</tr>
<tr>
<td>Automation level</td>
<td>Semi-automatic</td>
<td>Automatic</td>
</tr>
</tbody>
</table>

5.4 User Interfaces

<table>
<thead>
<tr>
<th>Type</th>
<th>List of Publications</th>
<th>Metadata of publication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Provides to the User the capability to explore the list of publications that is connected to a specific publication.</td>
<td>Provides to the User all extracted metadata and figures from a selected publication along</td>
</tr>
</tbody>
</table>
### 5.5 Data Processing Scenarios

<table>
<thead>
<tr>
<th>No</th>
<th>Type</th>
<th>Action Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Communication</td>
<td>Retrieve from AGRIS JSON metadata of scientific publications as well as the actual publication in PDF format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Retrieve from CORE JSON metadata of scientific publications as well as the actual publication in PDF format.</td>
</tr>
<tr>
<td>1.2</td>
<td>Storage / Publish</td>
<td>The metadata are stored in the Relational Database Management System (RDBMS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The PDF documents are stored in the filesystem of the underlying operating system.</td>
</tr>
<tr>
<td>2.1</td>
<td>Text-mining</td>
<td>Extract the textual transcript from each PDF Document using the PDF Extractor</td>
</tr>
<tr>
<td>2.2</td>
<td>Workflow</td>
<td>Associate PDF documents with Grape Variety terms using the Grape Varieties Extractor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[if associated] extract AgroVoc terms using the AgroVoc Extractor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[if associated] extract Figures (and related captions) using the PDF Extractor</td>
</tr>
<tr>
<td>2.3</td>
<td>Storage / Publish</td>
<td>The metadata are stored in the Relational Database Management System (RDBMS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The extracted images are stored in the filesystem of the underlying operating system.</td>
</tr>
</tbody>
</table>

### 5.6 Limitations

<table>
<thead>
<tr>
<th>Potential Limitation</th>
<th>Mitigation Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scalability:</strong></td>
<td>The current storage capacity is sufficient for the time being. However, due to the rich bibliographic data sources that have been selected, the storage capacity may be exhausted in the future. In this case, GRNET will increase of the storage capacity of the hosting environment.</td>
</tr>
</tbody>
</table>
5.7 Release Plan

There are 3 key releases for the application components, as follows:

- Jan 2017 - 1st Application Components Release
- August 2017 - 2nd Application Components Release
- Feb 2018 - 3rd Application Components Release

We expect the following parts of the OpenMinTeD infrastructure to be available at each release:

- Jan 2017 - Registry Available
- August 2017 - Initial version of workflow service available
- Feb 2018 - Annotation viewing + editing service available

Consequently, the release plan for these application components will be as follows:

Jan 2017

We will deliver the application components, as autonomous - stand-alone – components / services. The components will be registered in the OpenMinTeD registry.

August 2017

We will deliver the application as a self-contained dockerised workflow registered on the OpenMinTeD infrastructure. The Docker container will follow the specification produced by WG4, ensuring that it follows a common interface. The application will be placed in the registry to allow its usage as part of the OpenMinTeD platform.

Feb 2018

For the third and final application release, the application will be released as configurable workflow. Each component in an application will be placed into a Docker image and the OpenMinTeD workflow editor will be used to configure the workflow. This will allow new users to change the functionality of the workflow to suit their needs, as well as to create new workflows from the building blocks of other workflows.
6. **AS-B: Text-mining over RSS Feeds: Food Safety & Water Health**

The text-mining problem to be addressed in this Use Case is the extraction of geolocation information from unstructured RSS feeds. Two main communities are targeted:

- **Food Safety Community.** Via the network of 56 organisations participating in the Global Food Safety Partnership (GFSP, [http://gfsp.agroknow.com/](http://gfsp.agroknow.com/)), the need for a text-mining pipeline capable of extracting geolocation information over unstructured food safety related news (e.g. about foodborne illness outbreaks, food recalls etc.) was highlighted. GFSP is a unique public-private initiative dedicated to improving the safety of food in middle-income and developing countries. They bring together fishers and farmers, business and industry, governments, regulatory bodies, international development organizations, and civil society to drive a globally-coordinated and locally-driven food safety approach.

- **Water Health Community.** The same need was also identified by the Global Knowledge Hub for Water Pathogens ([www.waterpathogens.org](http://www.waterpathogens.org)), a community of 150 organisations working on water quality, which is coordinated by The Michigan State University (MSU, [https://msu.edu](https://msu.edu)). MSU has established the MSU Water Science Network to advance innovative science that addresses the most important water problems facing society.

Two main ontological resources will be used for assigning location URIs, namely, Geonames ([www.geonames.org](http://www.geonames.org)) and the FAO Geopolitical Ontology ([www.fao.org/countryprofiles/geoinfo/en/](http://www.fao.org/countryprofiles/geoinfo/en/)).

The front-end application (for both targeted communities) will be a web-based viewer (for mobile devices) capable of projecting relevant (to each community) information per selected country/region.

### 6.1 Resources

#### 6.1.1 Document Formats

<table>
<thead>
<tr>
<th>Name</th>
<th>Specification</th>
<th>Intended Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extensible Markup Language (XML)</td>
<td><a href="https://www.w3.org/TR/xml/">https://www.w3.org/TR/xml/</a></td>
<td>Text Mining</td>
</tr>
</tbody>
</table>

#### 6.1.2 Knowledge Bases

<table>
<thead>
<tr>
<th>Geonames</th>
<th>FAO Geopolitical Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>3.1</td>
</tr>
</tbody>
</table>
The selection of suitable RSS Feed sources will be finalised by January 2017. Indicative sources include:

<table>
<thead>
<tr>
<th></th>
<th>Food Safety News</th>
<th>WaterWorld</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>1.81</td>
<td>-</td>
</tr>
<tr>
<td><strong>Link to documentation</strong></td>
<td>-</td>
<td><a href="http://www.waterworld.com/rss.html">http://www.waterworld.com/rss.html</a></td>
</tr>
<tr>
<td><strong>Intended use</strong></td>
<td>RSS Feed Source</td>
<td>RSS Feed Source</td>
</tr>
<tr>
<td><strong>Data Format</strong></td>
<td>XML</td>
<td>XML</td>
</tr>
<tr>
<td><strong>Approx size</strong></td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### 6.1.3 Tools/Components

<table>
<thead>
<tr>
<th></th>
<th>GeoNames Extractor</th>
<th>Geopolitical Extractor</th>
<th>RSS Feed Reader</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>1.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Licence</strong></td>
<td>GPL 3.0</td>
<td>GPL 3.0</td>
<td>GPL 3.0</td>
</tr>
<tr>
<td><strong>Link to documentation</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
### Link to Code

<table>
<thead>
<tr>
<th>Task Performed</th>
<th>I/O Format</th>
<th>Deployment Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search and Rank Geolocation Terms in PDF documents</td>
<td>JSON/XML</td>
<td>Standalone</td>
</tr>
<tr>
<td>Search Geolocation terms in the FAO Geopolitical Ontology</td>
<td>RDF</td>
<td>Standalone</td>
</tr>
<tr>
<td>Ingests RSS Feeds from specified sources</td>
<td>XML</td>
<td>Standalone</td>
</tr>
</tbody>
</table>

### System Requirements

- a) Java Virtual Machine 1.7
- b) Lucene 6.2.1

### Performance Requirements

- -

### Static Resources

- GeoNames 3.1
- Geopolitical Ontology 1.1

### 6.1.4 Services

**GeoNames Webservice**

<table>
<thead>
<tr>
<th>Address</th>
<th>Search GeoNames Ontology for geolocation terms</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://api.geonames.org/">http://api.geonames.org/</a></td>
<td></td>
</tr>
<tr>
<td>I/O Format</td>
<td>STRING/ JSON</td>
</tr>
<tr>
<td>Terms of Use</td>
<td>CC BY 3.0</td>
</tr>
<tr>
<td>Hosting Institution</td>
<td>Geonames</td>
</tr>
<tr>
<td>Interface Type</td>
<td>Rest</td>
</tr>
</tbody>
</table>

### 6.2 Deployment Plan

The described components (GeoNames Extractor, Geopolitical Extractor and RSS Feed Reader) will be deployed as standalone tool in a single virtual machine (VM). The required infrastructure is provided by the Greek Research and Technology Network (https://grnet.gr/en/) through the cloud service OKEANOS (https://okeanos.grnet.gr/home/).

The resources retrieved by the RSS Feed sources will be stored as XML documents in the same VM.
6.3 Data Interfaces

<table>
<thead>
<tr>
<th>RSS Feed Reader</th>
<th>News Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Type</td>
<td>Data Stream</td>
</tr>
<tr>
<td>Protocol</td>
<td>REST</td>
</tr>
<tr>
<td>Data transfer format</td>
<td>XML</td>
</tr>
<tr>
<td>Automation level</td>
<td>Automatic</td>
</tr>
</tbody>
</table>

6.4 User Interfaces

<table>
<thead>
<tr>
<th>List of RSS Feeds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Indicative Information</td>
</tr>
</tbody>
</table>

6.5 Data Processing Scenarios

<table>
<thead>
<tr>
<th>No</th>
<th>Type</th>
<th>Action Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data Collection</td>
<td>Retrieve RSS Feeds from Food Safety.</td>
</tr>
<tr>
<td></td>
<td>Communication</td>
<td>Retrieve RSS Feeds from WaterWorld.</td>
</tr>
</tbody>
</table>
1.2 Storage / Publish

The metadata are stored in the Relational Database Management System (RDBMS) and the RSS Feeds are stored as XML documents in the filesystem of the underlying operating system.

2. Data Processing

2.1 Text-mining

Extract the Geolocations from the RSS Feeds using the GeoNames Extractor.

2.2 Workflow

Associate the extracted Geolocation with the FAO Geopolitical Ontology.

2.3 Storage / Publish

- The metadata are stored in the Relational Database Management System (RDBMS).
- [if associated] extract Figures (and related captions) using the PDF Extractor.
- The extracted images are stored in the file system of the underlying operating system.

6.6 Limitations

<table>
<thead>
<tr>
<th>Potential Limitation</th>
<th>Mitigation Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scalability:</strong> The basic limitation is posed by the storage capacity of the hosting environment.</td>
<td>The current storage capacity is sufficient for the time being. However, due to the rich bibliographic data sources that have been selected, the storage capacity may be exhausted in the future. In this case, GRNET will increase the storage capacity of the hosting environment.</td>
</tr>
</tbody>
</table>

6.7 Release Plan

There are 3 key releases for the application components, as follows:

- **Jan 2017** - 1st Application Components Release
- **August 2017** - 2nd Application Components Release
- **Feb 2018** - 3rd Application Components Release

We expect the following parts of the OpenMinTeD infrastructure to be available at each release:

- **Jan 2017** - Registry Available
- **August 2017** - Initial version of workflow service available
- **Feb 2018** - Annotation viewing + editing service available
Consequently, the release plan for these application components will be as follows:

**Jan 2017**

We will deliver the application components, as autonomous - stand-alone – components / services. The components will be registered in the OpenMinTeD registry.

**August 2017**

We will deliver the application as a self-contained dockerised workflow registered on the OpenMinTeD infrastructure. The Docker container will follow the specification produced by WG4, ensuring that it follows a common interface. The application will be placed in the registry to allow its usage as part of the OpenMinTeD platform.

**Feb 2018**

For the third and final application release, the application will be released as configurable workflow. Each component in an application will be placed into a Docker image and the OpenMinTeD workflow editor will be used to configure the workflow. This will allow new users to change the functionality of the workflow to suit their needs, as well as to create new workflows from the building blocks of other workflows.
7. **AS-C: Microbial Biodiversity**

This chapter contains the detailed specification for the design of the application "Microbial Biodiversity". The document provides the information that is required by the implementation, testing and maintenance of the application. Its purpose is also to anticipate the reuse and reconfiguration of the AS-C workflow by OMTD users by providing description of all necessary components including ancillary tools.

It has been written as a stand-alone document that can be read in isolation and used as such by application designers. It contains some overlapping material with Chapter 6 and 7 that has been replicated deliberately for completeness.

The Microbial Biodiversity text-mining application is part of a parent application that is currently under development at the IFB (French Bioinformatics Institute\(^1\)). The text-mining workflow is composed of four main stages. It first harvests and converts into text, scientific papers and free-text fields of genomics databases. Relevant documents selection is done using shallow criteria (such as lists of keywords) and manual review of relevant journals.

Natural Language Processing (NLP) steps then pre-process documents, namely tokenization, sentence segmentation, lemmatization, POS tagging and syntactic parsing. These analyses will serve as resources for subsequent and more complex IE tasks.

Next, entity recognition and categorization are performed to detect relevant entities (e.g. bacteria, habitats and phenotypes) from the documents and map them to normalized knowledge bases (e.g. taxonomies, ontologies). It uses a combination of NLP and Machine Learning (ML) methods trained on a reference corpus that has been manually annotated by food microbiology experts. In a next step, relations between tagged entities are extracted, based on machine learning.

Finally, the TDM workflow results are used by the target bioinformatics application hosted by the Migale\(^2\) platform of the French Bioinformatics Infrastructure IFB. Beyond extracted information, the TDM workflow share three types of information with the bioinformatics application. They are knowledge base references (e.g. taxa), the metadata for textual information display (e.g. snippet) and the user feedback. The first type of information, knowledge base references, requires synchronization of the updates between the TDM pipeline and the parent application since it comes from external resources.

The application will be deployed using the OpenMinTeD infrastructure. The necessary components from Alvis are being converted to be compatible with OpenMinTeD.

---

1. [https://www.france-bioinformatique.fr/en](https://www.france-bioinformatique.fr/en)
7.1 Resources

The application uses two kinds of document corpora, (1) corpora for training the machine learning methods that will be used for IE prediction (Bacteria Biotope and Florilège) and (2) a production corpus called FoodBiodiv from which the TDM pipeline extracts the relevant information.

7.1.1 Document Formats

<table>
<thead>
<tr>
<th>Name</th>
<th>Specification</th>
<th>Intended Use</th>
<th>Licence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Florilège corpus</td>
<td>See Appendix 1</td>
<td>Training of ML for IE. Manual reference annotation is achieved by a team of fifteen microbiologists.</td>
<td>CC-BY-SA license v3.0 (INRA)</td>
</tr>
<tr>
<td>Bacteria Biotope corpus</td>
<td>BioNLP-ST’11(^3) [Bossy et al., 2012], BioNLP-ST’13(^4) [Bossy et al., 2015], BioNLP-ST’16(^5) [Deléger et al., 2016]</td>
<td>Training of ML for IE. Manual reference annotation has been achieved by biology and computer science experts (see [Deléger et al., 2016] for details on the reference annotation).</td>
<td>CC-BY-SA license v3.0. (INRA)</td>
</tr>
<tr>
<td>Food Microbial Biodiversity corpus</td>
<td>See Appendix 1</td>
<td>Extraction of relevant information. Periodic update of the corpus with new publications that fit the criteria.</td>
<td>when known, formal description of licenses are stored.</td>
</tr>
<tr>
<td>Text Food Microbial Biodiversity corpus (TFoodBiodiv)</td>
<td>-</td>
<td>Extraction of relevant information. Periodic update of the corpus with new publications that fit the criteria.</td>
<td></td>
</tr>
</tbody>
</table>

7.1.2 Knowledge Bases

<table>
<thead>
<tr>
<th>NCBI taxonomy</th>
<th>OntoBiotope</th>
<th>Latin expression lexicon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Not Versioned</td>
<td>The current public version is BioNLP-ST’16</td>
</tr>
</tbody>
</table>

\(^3\) http://2011.bionlp-st.org/home/bacteria-biotopes
\(^4\) http://2013.bionlp-st.org/tasks/bacteria-biotopes
\(^5\) http://2016.bionlp-st.org/tasks/bb2
### 7.1.3 Tools/Components

The tool descriptions are gathered in functional categories that represent the main steps of the workflow. In most case the link to the code will be given in future versions of this document.

#### 7.1.3.1 Corpus Construction Tools

The diagram Corpus Builder (in the appendices) shows the role of the corpus construction tools.

**Extract DOIs**

- **Performed task**: read scientific literature records and extracts information on DOIs from them, removes DOI already present in the corpus
- **Version**: To be developed
- **Input format**: bibliographic database (WoS or PubMed)
- **Output formats**: XML

**Analyse publisher landing page**

- **Performed task**: check if the landing web page is correct, if not it searches for the URL of the full-paper. It outputs the number of successes.
The diagram Corpus Converter (in the appendices) shows the role of the corpus converter tools.

**XML Reader (XMLReader2)**
- Performed task: Converts documents from publishers in XML/HTML format (Corpus Converter).
- Version: 0.5rc
- License: ALv2
- Input format: XML to OMTD internal representation
- Deployment mode: AlvisNLP/ML Module
- Requirements for system environment: AlvisNLP/ML
- Performance requirements and limits: RAM for corpus size
- Used static resources and models: the structure of the input XML is handled through the xlsTransform XSLT stylesheet.

**TextFileReader**
- Performed task: Converts documents from publishers in text format (Corpus Converter).
- Version: 0.5rc
- License: ALv2
- Input/Output formats: unicode text to AlvisNLP/ML internal representation
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters), AlvisNLP/ML
- Performance requirements and limits (used memory, processing speed, multi-threading): NA
- Used static resources and models: No

**PDF Reader (TikaReader)**
- Performed task: Converts documents from publishers in PDF format (Corpus Converter).
- Version: 0.5rc (requires Tika 1.13)
- License: ALv2
- Input/Output formats: PDF to AlvisNLP/ML internal representation
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters), AlvisNLP/ML
- Performance requirements and limits (used memory, processing speed, multi-threading): NA
- Used static resources and models: No

7.1.3.2 Corpus Preprocessing Tools

**WoSMig**
- Version: 0.5rc
- License: ALv2
Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/WoSMig/doc
Performed task: Word segmentation (Tokenization) in Segmenter-POS-Tagger
Input/Output formats: AlvisNLP/ML internal representation
Deployment mode: AlvisNLP/ML module
Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
Used static resources and models: none

SeSMig
- Version: 0.5rc
- License: ALv2
- Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/SeSMig/doc
- Performed task: Sentence segmentation in Segmenter-POS-Tagger
- Input/Output formats: Alvis internal representation
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
- Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
- Used static resources and models: none

TreeTagger
- Version: 0.5rc (relies on tree-tagger version 3.2)
- License: ALv2 (for AlvisNLP/ML). TreeTagger 3.2: “freely available for research, education and evaluation. Usage of the system for commercial purposes is forbidden.” (TreeTagger License terms\(^6\))
- Link to documentation: http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/ (original doc)
  http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/TreeTagger/doc (AlvisNLP/ML doc)
- Link to the code: Executable code (not open source) available at http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/
- Performed task: Part-of-Speech tagging and lemmatization in Segmenter-POS-Tagger
- Input/Output formats: AlvisNLP/ML internal representation
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML, Linux/Mac-OS/Windows
- Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
- Used static resources and models: Parameter files available for several languages; The English parameter file is used in this use case.

\(^6\)http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/Licence
Genia Tagger

- Version: 0.5rc (relies on Genia Tagger version 3.0.2)
- License: ALv2 (for AlvisNLP/ML); Genia Tagger 3.0.2: “Redistribution and use in source and binary forms, with or without modification, are permitted for non-commercial purposes.”
- Link to documentation: http://www.nactem.ac.uk/GENIA/tagger/ (original doc)
- Link to the code: http://www.nactem.ac.uk/tsujii/GENIA/tagger/geniatagger-3.0.2.tar.gz
- Performed task: Part-of-Speech tagging and lemmatization in Segmenter-POS-Tagger
- Input/Output formats: Alvis internal representation
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML, Linux, gcc compiler
- Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
- Used static resources and models: POS-tag/chunk/EN models

7.1.3.3 Entity Recognition and Normalisation Tools

StanfordNER

- Version: 0.5rc (uses StanfordNER 3.4)
- License: ALv2 (for AlvisNLP/ML) and GNU General Public License (v2 or later) (for StanfordNER)
- Link to documentation: http://nlp.stanford.edu/software/CRF-NER.shtml (original doc)
- Performed task: Named entity recognition (specifically, Geographical entities in AS-C NER and Normalizer)
- Input/Output formats: AlvisNLP/ML internal representation
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters) AlvisNLP/ML and StanfordNER
- Performance requirements and limits (used memory, processing speed, multi-threading) NA
- Used static resources and models: NER model

ChemSpot

- Version: version 2.0
- License: Common Public License 1.0
- Link to documentation: https://www.informatik.hu-berlin.de/de/forschung/gebiete/wbi/resources/chemspot/chemspot
- Link to the code: https://github.com/rockt/ChemSpot
- Performed task: Chemical entity tagging (to be used in AS-C NER and Normalizer)
- Input/Output formats: raw text file (input), tabular text file (output)
- Deployment mode: standalone tool or AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters)
• Performance requirements and limits (used memory, processing speed, multi-threading): 16G RAM
• Used static resources and models: tagger model, dictionary, id list.

Chemical Entity Recogniser (Component in Argo)
• Version: 0.1-SNAPSHOT
• License: NaCTeM Proprietary Licence
• Link to documentation: None
• Link to the code: Not available outside of NaCTeM’s servers
• Performed task: recognising metabolites in biomedical text
• Input/Output formats: UIMA CAS - Argo type system
• Deployment mode: workflow component
• Requirements for system environment (OS, installed libs, language interpreters): UIMAcompatible
• Performance requirements and limits (used memory, processing speed, multi-threading): Unknown
• Used static resources and models: None

SimpleProjector2
• Version: 0.5rc
• License: ALv2
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/SimpleProjector2/doc
• Performed task: Lexicon projection. Used in AS-C NER and Normalizer
• AlvisNLP/ML internal representation format
• Vocabulary format: tabular text file (one entry per line)
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for
dictionary and corpus size
• Used static resources and models: may be used with any lexicon (in the required format)

RegExp
• Version: 0.5rc
• License: ALv2
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/RegExp/doc
• Performed task: Matches a regular expression on sections contents and create an annotation for
each match. Used in AS-C NER and Normalizer
• Input/Output formats: AlvisNLP/ML internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for
corpus size
• Used static resources and models: Regular expressions

**PatternMatcher**

• Version: 0.5rc
• License: ALv2
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/PatternMatcher/doc
• Performed task: Matches a specific Pattern (with previous annotation) on sections contents and create an annotation for each match. Used in AS-C NER and Normalizer
• Input/Output formats: Alvis internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: Regular expression-like patterns

**YateaExtractor**

• Version: 0.5rc (relies on BioYaTeA CPAN module version 0.11)
• License: ALv2 (for AlvisNLP/ML) and The Perl 5 License (Artistic 1 & GPL 1)” (for BioYatea CPAN module)
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/YateaExtractor/doc
• Performed task: Term extraction (using the BioYaTeA tool) in AS-C NER and Normalizer
• Input/Output formats: Alvis internal representation, extracted terms in XML file
• Deployment mode: AlvisNLP/ML module, CPAN module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML, CPAN
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: term extraction patterns

**ToMapProjector**

• Version: 0.5rc
• License: ALv2
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/TomapProjector/doc
• Performed task: Normalize entities according to an ontology or terminology (used in AS-C NER and Normalizer)

---

7 http://dev.perl.org/licenses/
• Input/Output formats: Alvis internal representation, XML serialized lexicon analysis, YaTeA extracted terms as XML
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, and limits (used memory, processing speed, multi-threading): RAM for corpus and lexicon size
• Used static resources and models: term extraction, pre-analyzed lexicon/ontology

**ToMapTrain**
• Version: 0.5rc
• License: ALv2
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/ToMapTrain/doc
• Performed task: Process the terms of an ontology/lexicon in order to use them with the normalization module (ToMapProjector). Used in Ontology/Lexicon Analyzer
• Input/Output formats: Alvis internal representation, XML serialized lexicon analysis
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, and limits (used memory, processing speed, multi-threading): RAM for corpus and lexicon size
• Used static resources and models: ontology/lexicon to be analyzed

7.1.3.4 Training and Relation Extraction Tools

**CCG parser**
• Version: 0.5rc (relies on CCG Parser 1.00)
• License: ALv2
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/CCGParser/doc
• Performed task: Dependency parsing.
• Input/Output formats: AlvisNLP/ML internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: language models

**TEES trainer**
• Version: 2.0.1
• License: GNU General Public License
• Link to documentation: https://github.com/jbjorne/TEES/wiki/Training
• Link to the code: https://github.com/jbjorne/TEES
• Performed task: Train a corpus and produce a SVM-based learning model
• Input/Output formats:
  o Interaction XML
BioNLP ST format

- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters)
  - System: Unix/Lunix
  - Language: Python
  - Dependencies: Python libs
- Performance requirements and limits (used memory, processing speed, multi-threading): RAM for training data set (esp. negative examples)
- Used static resources and models: Training and test corpora, feature definitions

TEES classifier

- Version: 2.0.1
- License: GNU General Public License
- Link to documentation: https://github.com/jbjorne/TEES/wiki/Training
- Link to the code: https://github.com/jbjorne/TEES
- Performed task: Predict (classify) relations from a corpus and SVM-based learning model
- Input/Output formats:
  - Interaction XML
  - BioNLP ST format
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters)
  - System: Unix/Lunix
  - Language: Python
  - Dependencies: Python libs
- Performance requirements and limits (used memory, processing speed, multi-threading)
- Used static resources and models: SVM-based Learning Model

BioNLP format converter

- Version: 2.0.1
- License: GNU General Public License
- Link to documentation: https://github.com/jbjorne/TEES/wiki/The-Preprocessor
- Link to the code: https://github.com/jbjorne/TEES/tree/master/Utils
- Performed task: Conversion between BioNLP ST format and Interaction XML
- Input/Output formats:
  - BioNLP ST format
  - Interaction XML
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters)
  - System: Unix/Lunix
D9.2 – Community Driven Applications Design Report

- Language: Python
- Dependency resolutions: Python libs
  - Performance requirements and limits (used memory, processing speed, multi-threading)
  - Used static resources and models: None

### 7.1.3.5 Training corpus design

#### Preannotation
- Performed task: preannotates the documents with respect to the annotation schema.
- Version: Alvis
- Input format: XML internal OMTD format
- Output formats: XML internal OMTD format

#### AlvisAE converter
- Performed task: convert documents XML internal OMTD format in AlvisAE JSON format.
- Version: Alvis
- Input format: XML internal OMTD format
- Output formats: AlvisAE JSON format

#### AlvisAE
- Performed task: Manual annotation.
- Version: Alvis
- Input format:
  - documents: AlvisAE JSON format,
  - annotation schema: JSON,
  - Campaign information (people, tasks) in AlvisAE JSON format
- Output formats: documents, in AlvisAE JSON format

#### Annotation Progress Manager
- Performed task: Display annotation progress per person, per task and per document.
- Version: Alvis
- Input format: AlvisAE campaign information format
- Output formats: N/A

#### AlvisAE export
- Performed task: export AlvisAE annotations in RDF, XML and BioNLP-ST formats
- Version: Alvis
- Input format: AlvisAE information
- Output formats: RDF, XML and BioNLP-ST formats

### 7.1.3.6 Tools for communication between TDM and Florilège parent application

#### Annotation Export
• Version: Alvis
• Performed task: Transform annotations in OMTD format into annotations in the external OMTD-Florilège format as a file.
• Input formats: Alvis/OMTD internal format
• Output formats: external OMTD-Florilège format
• Deployment mode: Workflow component

Florilège KB export
• Version: to be developed
• Performed task: build an archive into the OMTD-Florilège format of the updated versions of the KB maintained by MaIAGE. To be used by Florilège for indexing the annotations.
• Input formats: OMTD internal format
• Output formats: external OMTD-Florilège format
• Deployment mode: workflow component

Florilège KB import
• Version: to be developed
• Performed task: convert the archive of the updated versions of the KB received from Florilège into OMT KB internal format.
• Input formats: external OMTD-Florilège format
• Output formats: OMTD internal format
• Deployment mode: Workflow component

7.1.4 Services
NCBI Taxonomy download
• Performed Tasks: retrieve the last version of the NCBI Taxonomy
• Input formats: NA
• Output formats: tar, gz, proprietary
• Terms of use (esp. performance limitations): https://uts.nlm.nih.gov/license.html free of use, not OA/OSS
  Hosting institution: NCBI
• Type of interface: FTP download

Elsevier API
• Address: http://dev.elsevier.com/
• Performed Tasks: get full text files from DOIs
• Input formats: API query with a list of DOIs
• Output formats: XML or HTML
• Terms of use (esp. performance limitations): (source: policy 8 for TDM use cases) “We have adopted a license-based approach which enables researchers at subscribing institutions to register for an API key to text mine for non-commercial research purposes and to gain access to full text content in XML for this purpose.” “In addition to supporting researchers at subscribing institutions to text mine, our policy also allows the following: Open access content: Everyone can download open access content to mine via our Full Text API, in line with the individual article’s user license.”

Performance limitations: “there are no hard limits on the number of items that may be downloaded via our API. Nevertheless, a reasonable and customary rate limit remains in place to ensure equal access to the API for all users, and we continue to ask users to use our service responsibly.”

• Hosting institution: Elsevier
• Type of interface: RESTful

Springer API
• Address: https://dev.springer.com/
• Performed Tasks: get full text files from DOIs
• Input formats: API query with a list of DOIs
• Output formats: PDF
• Terms of use (esp. performance limitations): (source: policy 9) “Springer grants text- and data-mining rights to subscribed content to researchers via their institutions, provided the purpose is non-commercial research.” “Publications or analyses resulting from TDM of subscribed content may include quotations from the original text of up to 200 characters, or 20 words, or 1 complete sentence. They should cite the original Springer content in the form of a DOI link. Permission to reproduce images may be granted on a case-by-case basis. For Open Access (OA) publications from Springer, BioMed Central and SpringerOpen, TDM is usually allowed without restrictions since the majority of our OA content is licensed under CC-BY.”

Performance limitations: “TDM researchers are requested to be considerate and limit their downloading speed to a reasonable rate.”

• Hosting institution: Springer
• Type of interface: RESTful

PMC OA Web Service
• Address: https://www.ncbi.nlm.nih.gov/pmc/utils/oa/oa.fcgi
• Performed Tasks: get bibliographic metadata and document URL full text files from DOIs
• Input formats: doi
• Output formats: PDF
• Terms of use (esp. performance limitations): (source: Developer resources 10) “The use of our APIs is entirely free, and doesn’t require an API key”. “Include two parameters that help to identify

---

8 https://www.elsevier.com/about/company-information/policies/text-and-data-mining
10 https://www.ncbi.nlm.nih.gov/pmc/tools/developers/
your service or application to our servers: tool should be the name of the application, as a string value with no internal spaces, and email should be the e-mail address of the maintainer of the tool, and should be a valid e-mail address."

- Performance limitations: “Do not make concurrent requests, even at off-peak times"
- Hosting institution: NCBI
- Type of interface: REST

**Digital Object Identifier System**

- Address: https://www.doi.org
- Performed Task: get URL from DOIs
- Input formats: URL query: composed of https://doi.org/followed by DOI number. e.g. https://doi.org/10.1007/978-3-319-13674-5_28
- Output format: URL of the landing page
- Hosting institution: DOI Foundation
- Type of interface: http

**Crossref**

- Address: http://www.crossref.org
- Performed Task: get URL (resource field) and bibliographic metadata from DOIs
- Input formats: Open URL query or http query
- Output format: xsd_xml or UNIXREF format. See http://help.crossref.org/doi-to-metadata-query
- Terms of use (esp. performance limitations): (source: http://www.crossref.org/01company/free_services_agreement.html) “CrossRef asserts no claims of ownership to individual items of bibliographic metadata and associated Digital Object Identifiers (DOIs) acquired through the use of the CrossRef Free Services. Individual items of bibliographic metadata and associated DOIs may be cached and incorporated into the user’s content and systems. […]”
- Best practice: use XML
- Performance limitations: Restricted to registered users
- Hosting institution: Crossref
- Type of interface: REST

### 7.2 Deployment Plan

The application is deployed on Migale infrastructure. Migale is a certified bioinformatics service platform that provides web application hosting, RDBMS hosting, virtualization, storage, and a cluster with more than 1000 cores for heavy data processing. The initial deployment (1st release) will use INRA TDM software infrastructure, Alvis Suite, including:

- AlvisNLP/ML: corpus processing, document format conversion, and model training.
- AlvisAE: training corpus annotation.
- AlvisIR: semantic search engine.
This initial deployment will handle only the Application Design and part of the Production Processing scenarios defined below. Indeed update scenarios will rely on features of the future OMTD infrastructure.

The redeployment on the OMTD infrastructure will take place gradually in several steps detailed below.

**Registry entries**

The most critical and reusable components and resources will have an entry in the OMTD registry.

**Alvis components as OMTD components**

Alvis software components will be wrapped as OMTD components so that they can be included in OMTD workflows. The integration of components will take place in several progressive steps.

AlvisNLP/ML is the first component to be integrated. It can serve as a generic corpus processing component. The integration of AlvisNLP/ML will allow to redeploy most of the automatic corpus processes described in this document on the OMTD infrastructure.

Individual components of AlvisNLP/ML may be deployed individually so that other workflows can take advantage of their functionality. The individually redeployed components will be selected according to their reusability by other partners and use-cases and originality.

The manpower necessary for wrapping AlvisNLP/ML and its components depends on:

- the API provided by OMTD for wrapping components,
- how much the OMTD data interface is compatible to AlvisNLP/ML data model,
- the amount of OMTD-SHARE metadata input will be required.

As an annotation editor, AlvisAE will be integrated through the AERO protocol (OMTD internal document) which allows OMTD to import and export data from and to annotation editors, as well as to control annotation projects.

**Integration in the host application**

One of the greatest benefits anticipated from the OMTD infrastructure is the ability to build workflows that can communicate with external applications, the host application in this use-case. The most appealing features are:

- The management of different versions of resources and components; the ability to chose the version for each.
- The querying of the environment of a workflow: which resources are used? Which elements have a newer version?
- The querying of metadata associated to different elements, in particular, for annotated corpora, the details of the processing environment (resource versions for instance).
• The programmatic control and monitor of workflow operations: the host application will need to be able to launch corpus processing, monitor its advancement and download the result with as few human interventions as possible.

• Logging facilities and error handling: how easily the host application manager can monitor the progress of the corpus processing, be notified of errors, and take action about errors.

Local OMTD deployment on Migale infrastructure
The local deployment of an OMTD on the Migale infrastructure will be considered in the following situations:

• The communication between the host application and OMTD entails the distant transmission of so much data that bandwidth becomes a bottleneck.

• This use case relies on the processing of closed access documents, which requires the use of a private space in OMTD. Even so the legality of distant processing of closed access documents might force us to deploy a local instance of OMTD.

The deployment of a local instance of OMTD would require a sizeable amount of work power that would be detrimental.

7.3 Data Interfaces
The AS-C application takes as input three types of knowledge bases (see format in Section 5.1.2)

• External resources (=NCBI taxonomy),

• KB maintained by MalAGE, the infrastructure that supports the TDM application (=OntoBiotope ontology, Latin expression lexicon)

• KB maintained by Migale, the infrastructure that supports the Florilège parent application

The two first sets of KB are exported by the TDM application to the parent application by the **KB export tool** (Section 5.1.3.6).

The latter is imported by the AS-C TDM application from the parent application by the **KB import tool** (Section 5.1.3.6).

The AS-C application also takes corpora as input (see format in Section 5.1.1)

• the initial corpus (FoodBioDiv) and its updates are handled by the Corpus Builder component

• the training corpora: Florilège and Bacteria Biotope.

The AS-C TDM application exports text annotations to the parent application. It is done by the **Annotation Export tool** (see format in section 5.1.3.6).

7.4 User Interfaces
The user interfaces in AS-C application are of two kinds:
• Interaction of knowledge manager and the domain expert with the reference annotation tools, AlvisAE and Annotation Progress Manager.
• Interaction of the end-user with the Florilège parent application: query, navigation, curation.

AlvisAE documentation includes the GUI for the project management and the end-user interactions. The main screens can be found in AlvisAE UI User Guide11.

The appendices gives an example of the Annotation Progression Manager GUI.

Mock-up screens of the Florilège parent application can be found in the appendices.

7.5 Data Processing Scenarios

The scenarios defined below refer to processes that are represented by diagrams in Annex 3.

There are three main scenarios that represent application life cycle phases:

• TDM application design: corpus design, including reference manual annotation, learning the classifier for prediction. Knowledge base download and conversion.
• TDM application production: production and export of annotations.
• TDM application update: curation, corpus and KB update, re-training..

(The detail below contains some overlapping material with Chapter 6 and 7 due to the similarity of scenarios, and that has been replicated deliberately for completeness of each part.)

Application Design

During the Application Design, OMTD supports automatic resource building (harvest, selection, and rewriting) and software component adaptation (mostly based on machine learning) that are necessary to the initial deployment of the application. At the end of this scenario, all the resources and components are ready for the first production processing of the corpus.

This is the scenario where we make the remaining operational choices with regards to the specifications. For instance if several software components were considered for a single step, then one of them is finally selected based on operational criteria: availability, ease of installation. The same approach is used to handle resource alternatives. The application manager in agreement with the host application manager will make these choices.

In this use-case, the most critical choices are:

• Dependency parsing: either CCG Parser, BBLIP, or Stanford
• Chemical NER: either ChemSpot, or Argo Chemical Tagger
• POS-tagging: tree-tagger, or genia tagger

The choices will be motivated by the availability of the software or resource, how easily they can be wrapped into OMTD, and by their performances (e.g. F-score, computation time). Each step of the automatic annotation will be evaluated separately. Metric evaluation will be used if a reference corpus is available for a particular task. Otherwise, end-users and the application manager will evaluate the automatic annotation by sampling the results and assessing their accuracy.

Currently we plan to use the following reference corpora:

- BioNLP-ST 2016 Bacteria Biotope corpus for NER and Normalization (Bacteria, Habitats), and Relation Extraction.
- Florilège corpus for NER and Normalization

One of the critical resources to build during the Application Design is the initial document set using the process described in (see Corpus Builder diagram). The result of the process is both the corpus that will be processed with its metadata, and the query that will be used to keep the corpus up to date. The corpus design involves a close cooperation between the TDM application manager, the parent application manager, and end-users that are domain-experts.

Finally, the Application Design scenario includes the acquisition of resources specific to this application including the design of training corpora. The training corpora annotation campaign involves knowledge managers and domain-experts (see Corpus Manual Annotation diagram).

**Production processing**

The Production Processing scenario represents the main lifecycle phase of the application. The processing of the corpus is triggered automatically and periodically. The resources, corpus, and components used are the most up-to-date at the time this scenario runs.

The Production scenario includes the automatic annotation of the corpus (AS-C Automatic Corpus Annotator diagram), then the transmission of the annotation to the host application (AS-C Data Exporter diagram).

The frequency of the Production processing scenario runs will depend on the duration of the processing of the whole corpus and the expected turnover of the resources i.e. KBs and the query result paper collection. This frequency may be adjusted during the lifecycle of the application. The Production processing update component is thus triggered by a timer like cron and the rest of the process is fully automatic.

It is expected that the Production process is incremental: it should only process new documents added to the corpus (see Resource Update scenario), unless one of the external resources is updated, in which case the whole corpus is re-processed.
Each time the update component triggers the Production processing scenario, it logs the specifics of the new version of the data. This log can be consulted and shall include the following information: a timestamp, the version of each component and resource, and the list of new references.

**Resource Update**

The Resource Update scenario is the set of procedures that ensure that the Production Processing uses the most up-to-date resources, and that both the TDM application and the host application refer to the same version of each resource. These procedures are automated as much as possible; however the distribution mode of some base material may require human overseeing. The Resource Update scenario involves diverse processes:

**Corpus increment**

The bibliographic query is re-run periodically in order to monitor new documents relevant to the application. Depending on the bibliographic database used for gathering references, this step may be automated. If not the TDM application manager will run the query and fetch the results. Nevertheless, the following steps for building the corpus (Corpus Builder) are triggered automatically.

**Acquisition of new versions of external resources**

The Florilège KB Import (See section 5.1.3.6, and 5.1.2 for resource reference) incorporates resources provided by external entities to the application. When these entities provide means to monitor updates, and to download automatically new versions, then this may be done automatically. However, some situations may require the validation of the application manager. For instance, major version shifts of resources may change the annotation results in such a way that they become unfamiliar to the end-users. Additionally changes in software behaviour may break the interoperability of the corpus annotation process.

Some resources have to be pre-processed and transformed in order to be used by software components of the corpus annotation (Ontology/Lexicon Analyzer, Lexicon Lemmatization, Relation Extraction Model Learner). These processes are automatically triggered upon the arrival (and the validation if necessary) of a new version of the base resource.

However, most of the resources are meant to be used for a single tool and they are often embedded to the tool distribution. In this case a new version of a resource might be exposed as new version of the tool itself. This is mostly the case for generic and linguistic resources (POS models, generic NE resources, etc.)

Finally, some resources are also used by the host application provided by Florilège KB export (section 5.1.3.6). Thus the host application must be informed when a new version of a resource is used and how this version can be accessed. This communication is automated in the same way as resource pre-processing.
7.6 Limitations

Access to the documents

The licensing and technical restrictions with regards to the automatic processing of some documents necessary to the use case are not clear yet. The whole corpus necessary to obtain useful data for the end-user may not be available. For instance the reference may not be available for direct download, or the harvester developed by OU might be barred from retrieving them.

To overcome this limitation the strategy adopted is to develop first resources that will maximize the amount of documents in the corpus.

NER Molecules

There are no available corpus annotated for chemical molecules in the domain of food microbiology, so the quality of existing molecule NER tools in this target domain is uncertain. We will integrate one or several state-of-the-art software based on availability for integration and on performances (evaluated on other corpora) but we might need a better quality component.

We expect to be able to reuse components integrated in OMTD for the use case LS-A that has a strong focus on chemical NER.

GRNET Infrastructure

As mentioned in section 3, legal or technical restrictions might force us to deploy another instance of OMTD on Migale’s infrastructure. In this case, the manpower necessary to the deployment will hinder the software integration of the use case into OMTD.

7.7 Release Plan

7.7.1 1st release (January)

OMTD deployment

- Entry of main components in the OMTD registry.
  - Restrictions: availability of metadata for components.

Application development (AlvisNLP/ML deployment)

- Scenarios: Application Design, Production Processing (initial processing)
- Corpus subset.
  - Restrictions: harvester development and harvest failures.
- Training corpus subset.
  - Restrictions: annotation campaign advancement.
- Prediction and categorization of taxa and habitats and relations between them, prediction of phenotypes.
  - Restriction: integration of relation recognition in AlvisNLP/ML and quality of predictions.
• Integration of manual annotations and TDM prediction in Migale Florilège database

7.7.2 2nd release (August 2017)
OMTD deployment
• Integration of AlvisNLP/ML in OMTD workflow engine as a Docker image.
• Storage of main resources in the OMTD infrastructure.
• Deployment of the Microbial Biodiversity TDM application on OMTD using a Docker image.
• Scenarios: Production Processing.
  o Restriction: the GRNET OMTD instance can be used.

Application development and deployment
• Query using terminology and ontology (hierarchy and synonyms) and display by Migale Florilège application
• Very large set of abstracts and titles of full paper corpus.
• Integration of external textual and non-textual data (DSMZ (BacDive database)) into Florilège application using web services

7.7.3 3rd release (January 2018)
OMTD deployment
• Entry of main components in the OMTD registry.
• Integration of the main individual AlvisNLP/ML components, and of resource processing tools (see section 5.1.3 Components/tools) using Docker images
• Scenarios: Resource Update
• Development of corpus design tools.
• Restriction: the GRNET OMTD instance can be used.

Application development
• Extension of corpus to free text field.
• Ontology display for query building.
• Integration with heterogeneous data (DSMZ, CIRMES), cont.
• Improvement of NER and relation extraction.
  o Restriction: training corpus availability.
• Full paper corpus.
  o Restriction: availability to the paper harvesters.
  o OMTD policy for non open access paper
• Curation and feedback tools.
  o Restriction: feasibility on the host infrastructure.
• Evaluation tools.
7.8 References


8. **AS-D: Linking Wheat Data With Literature**

This chapter contains the specification of the design of the application "Linking Wheat Data With Literature". The document provides the information that are required by the implementation, testing and maintenance of the application. Its purpose is also to anticipate the reuse and reconfiguration of the AS-D workflow by OMTD users by providing description of all necessary components including ancillary tools.

It has been written as a stand-alone document that can be read in isolation and used as such by application designers. It contains some overlapping material with Chapter 5 and 7 that has been replicated deliberately for completeness.

The “Linking Wheat Data With Literature” application is part of two parent applications, GnpIS\(^\text{12}\) and WheatIS search\(^\text{13}\), currently developed and maintained by the IFB\(^\text{14}\) Bioinformatics Infrastructure URGI INRA\(^\text{15}\). The text mining workflow processes scientific literature and textual fields of the GnpIS database in order to identify taxa, genes, markers and phenotypes that are described and accessible in both applications. The objective of the use case is to feed the database that supports the two parent applications with links between related in-house data and externally hosted texts.

The text-mining workflow is composed of four main stages.

- Relevant documents selection is done using shallow criteria (such as lists of keywords) and manual review of relevant journals by experts. Depending on sources, conversion from HTML or PDF to XML will be necessary.
- NLP steps will pre-process documents, namely tokenization, sentence segmentation, lemmatization, POS tagging and syntactic parsing. These analyses will serve as resources for subsequent and more complex IE tasks.
- Entity recognition and categorization will be performed to detect relevant entities (e.g. genes, taxa and phenotypes) from the documents and map them to normalized knowledge sources (e.g. taxonomies, ontologies) already in use in the parent applications. An existing corpus developed by INRA and seed companies within the FSOV SAM blé project [Nédellec et al. MTSR 2014] will be reused and curated for training NER methods and evaluation.
- The TDM results will be integrated by URGI in the database underlying the parent applications relying on entity identifiers alignments of genes, taxa, phenotypes and markers. The reference to the relevant scientific articles and snippets with entities and relations highlighted will be displayed by both GnpIS and WheatIS search applications. Regular updates will occur to either find new articles or find articles on new objects.

\(^{12}\) https://urgi.versailles.inra.fr/gnpis/
\(^{13}\) https://urgi.versailles.inra.fr/wheatis/
\(^{14}\) https://www.france-bioinformatique.fr/en
\(^{15}\) https://urgi.versailles.inra.fr/
8.1 Resources

8.1.1 Document Formats
The Wheat Corpus contains all relevant full-text papers for the Use Case that are technically and legally accessible. The formats of the documents depend on their source, which are diverse. They are detailed below.

Wheat corpus
- Link to specification/documentation: see Appendix 1
- Intended Use: Extraction of relevant information. Periodic update of the corpus with new publications that fit the criteria.
- Format: one format per publisher source, see below
- License: see below the license per publisher

XML Elsevier articles (part of Wheat corpus)
- Link to specification/documentation: Journal Article (JA) DTD 5.4.0\(^{16}\)
- Intended Use: Extraction of relevant information. Periodic update of the corpus with new publications that fit the criteria.
- License: Can be OA or under subscription. Formal description of licenses are stored.
- Format: XML, plus source metadata
- API: http://dev.elsevier.com/

HTML Springer articles (part of Wheat corpus)
- Link to specification/documentation: How to and policy\(^ {17}\)
- Intended Use: Extraction of relevant information. Periodic update of the corpus with new publications that fit the criteria.
- License: Can be OA or under subscription. Formal description of licenses are stored
- Format: HTML, plus source metadata
- API: https://dev.springer.com/

PDF Springer articles (part of Wheat corpus)
- Link to specification/documentation: How to and policy\(^ {17}\)
- Intended Use: Extraction of relevant information. Periodic update of the corpus with new publications that fit the criteria.
- License: Can be OA or under subscription. Apparently no formal description of licenses
- Format: PDF, plus source metadata
- API: https://dev.springer.com/

PDF PMC articles (part of Wheat corpus)
- Link to specification/documentation: https://www.ncbi.nlm.nih.gov/books/NBK3825/

---

\(^{16}\) https://www.elsevier.com/authors/author-schemas/elsevier-xml-dtds-and-transport-schemas

Intended Use: Extraction of relevant information. Periodic update of the corpus with new publications that fit the criteria.

License: OA

Format: PDF, plus source metadata

API: https://www.ncbi.nlm.nih.gov/pmc/utils/oa/oa.fcgi

SAM ble corpus

The SAM ble corpus is composed of 65 short extracts of full-papers from Springer journals.

Link to specification/documentation: See Appendices

Intended Use: Training of ML for IE. Manual reference annotation has been achieved by breeders and revised for OMT purposes. Also used for the evaluation of named entities extraction.


License: CC-BY-SA license v3.0

Text Wheat corpus

The Text Wheat Corpus contains all relevant full-text papers for the Use Case in text format. It is obtained by the conversion of the Wheat corpus into text (see above).

Link to specification/documentation:

Intended Use: Extraction of relevant information. The corpus will be periodically updated with new publications.

Format: OMTD

License: same as Wheat corpus.

GnplS DB Description Field Corpus

The GnplS DB Description Field Corpus contains excerpts of the GnplS database corresponding to the “description” field for genes and markers.

Link to specification/documentation: NA

Intended Use: Extraction of relevant information to enhance search results by allowing semantic search (hierarchies and synonyms). The corpus will be periodically updated with new DB entries.

Format: tabulated, identifier and content

8.1.2 Knowledge Bases

NCBI taxonomy

The NCBI Taxonomy is the tree of taxa for which at least one sequence has been submitted to GeneBank. This resource is maintained by the NCBI, it is curated, though the curation process is not clear. The taxonomy contains the taxa from the phyla to sub-specific ranks, the tree structure, and the most common synonyms.

Version: Not versioned.

• Intended use: recognition and normalization of the taxa mentioned in the text (AS-D NER and Normalizer)
• Data format: proprietary
• Approximate size (MBs, number of entities/classes): 36Mb compressed, ~1.4M taxa, 2M names
• Licence, terms of use: UMLS Metathesaurus License: http://uts.nlm.nih.gov/license.html

Wheat taxon lexicon
The wheat taxon list is provided by URGI that maintains it. Each entry includes a unique identifier, a preferred name and synonyms.

• Intended use: recognition and categorization of the taxa mentioned in the text (AS-D NER and Normalizer)
• Data format: tabulated text
• Approximate size (MBs, number of entities/classes): 23 entries

Gene lexicon
A gene list is provided by URGI that maintains it. Each entry includes a unique identifier, a preferred name and synonyms. It contains the names of the new complete genome sequence annotation in the form of TRAES_

• Intended use: indexing gene names in GnPiS and WheatIS. To be mapped to “Bibliome gene lexicon” for an integrated indexing of both experimental data of GnPiS and WheatIS and textual data (AS-D TDM workflow).
• Mapping to be achieved by Wheat genomics community in 2017 summer.

Bibliome gene lexicon
Internal resource
• Intended use: recognition and categorization of the gene names in the text (AS-D NER and Normalizer). The list contains the usual names as defined in gene catalogues (e.g. Lr26, Pm2).
• Data format: tabulated text
• Approximate size (MBs, number of entities/classes): 919 entries

Marker lexicon
The marker list is provided by URGI that maintains it. Each entry includes a unique identifier, a preferred name and synonyms.

• Intended use: recognition and categorization of the marker names in the text (AS-D NER and Normalizer)
• Data format: tabulated text
• Approximate size (MBs, number of entities/classes): 1290 entries

WIPO Wheat Phenotype Ontology
The WIPO Wheat Phenotype Ontology is the ontology used at URGI to reference the phenotypic experimental information stored and displayed in the GnPiS and WheatIS search applications. The
concepts of WIPO are individually linked to Wheat Trait Ontology (WTO) concepts when relevant in order to enable querying both experimental (WIPO index) and textual data (WTO index).

- **Version:** 1.0
- **Link to resource (url):** https://urgi-git.versailles.inra.fr/urgi-is/ontologies/tree/develop/Wheat
- **Intended use:** indexing wheat phenotype mentions in experimental data
- **Data format:** tabulated text, OWL to come soon, RDF to come soon.
- **Approximate size (MBs, number of entities/classes):** ~200 classes
- **Licence, terms of use:** CC-by or CC-by-nd 4.0 (TBD)

### Wheat Trait Ontology

The Wheat Trait Ontology defines wheat traits and phenotypes and environmental factors that impact wheat varieties. It is publically available on AgroPortal. The concept labels are close to the terms that are usually found in literature. The Use Case uses only the traits and phenotype part (called Plant Property). It is used in complement to the WIPO Wheat Phenotype Ontology.

- **Version:** 2.0a
- **Link to resource (url):** http://agroportal.lirmm.fr/ontologies/WHEATPHENOTYPE
- **Link to documentation:** http://link.springer.com/chapter/10.1007%2F978-3-319-13674-5_28
- **Intended use:** recognition and categorization of the wheat trait and phenotype mentions in the text (AS-D NER and Normalizer).
- **Data format:** obo
- **Approximate size (MBs, number of entities/classes):** 466 classes
- **Licence, terms of use:** CC-BY-SA license v3.0

### Latin expression lexicon

- **Version:** v1
- **Link to resource (url):** (part of AlvisNLP/ML)
- **Intended use:** linguistic preprocessing (word segmentation in Segmenter-POS-Tagger)
- **Data format:** .txt
- **Approximate size (MBs, number of entities/classes):** 37KB; 1889 entries

### StopWords

- **Version:** V1
- **Intended use:** Word Exclusion in the NER process
- **Data format:** text
- **Approximate size (MBs, number of entities/classes):** 179 words
- **Licence, terms of use:** CC-BY 2.0
8.1.3 Tools/Components
Some of the components are the same as the Microbial Biodiversity application (Chapter 6). Their descriptions are duplicated here for sake of completeness.

8.1.3.1 Corpus construction tools
The diagram Corpus Builder in Annex 3 shows the role of the corpus construction tools.

**Extract DOIs**
- Performed task: read scientific literature records and extracts information on DOIs from them, removes DOI already present in the corpus
- Version: To Be Developed
- Input format: bibliographic database (WoS or PubMed)
- Output formats: XML

**Analyse publisher landing page**
- Performed task: check if the landing web page is correct, if not it searches for the URL of the full-paper. It outputs the number of successes.
- Version: To Be Developed
- Input format: web page
- Output formats: URL

8.1.3.2 Corpus converting tools
The Corpus Converter diagram shows the role of the corpus converting tools.

**XML Reader** (XMLReader2)
- Performed task: Converts documents from publishers in XML/HTML format (Corpus Converter).
- Version: 0.5rc
- License: ALv2
- Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/XMLReader2/doc
- Input format: XML to OMTD internal representation
- Deployment mode: AlvisNLP/ML Module
- Requirements for system environment: AlvisNLP/ML
- Performance requirements and limits: RAM for corpus size
- Used static resources and models: the structure of the input XML is handled through a XSLT stylesheet.

**TextFileReader**
- Performed task: Converts documents from publishers in text format (Corpus Converter).
- Version: 0.5rc
- License: ALv2
PDF Reader (TikaReader)
- Performed task: Converts documents from publishers in PDF format (Corpus Converter).
- Version: 0.5rc (requires Apache Tika 1.13)
- License: ALv2
- Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/TikaReader/doc
- Input/Output formats: PDF to AlvisNLP/ML internal representation
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
- Performance requirements and limits (used memory, processing speed, multi-threading): NA
- Used static resources and models: No

8.1.3.3 Corpus preprocessing tools
The Segmenter-POS-Tagger diagram (see appendix 3) shows how those tools segment and tag the documents with linguistic information before NER recognition.

WoSMig
- Version: 0.5rc
- License: ALv2
- Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/WoSMig/doc
- Performed task: Word segmentation (Tokenization) in Segmenter-POS-Tagger
- Input/Output formats: AlvisNLP/ML internal representation
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
- Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
- Used static resources and models: none

SeSMig
- Version: 0.5rc
- License: ALv2
- Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/SeSMig/doc
- Performed task: Sentence segmentation in Segmenter-POS-Tagger
- Input/Output formats: Alvis internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: none

TreeTagger
• Version: 0.5rc (relies on tree-tagger version 3.2)
• License: ALv2 (for AlvisNLP/ML). TreeTagger 3.2: “freely available for research, education and evaluation. Usage of the system for commercial purposes is forbidden.” (TreeTagger License terms\textsuperscript{18})
• Link to documentation: http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/ (original doc)
  http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/TreeTagger/doc (AlvisNLP/ML doc)
• Link to the code: Executable code (not open source) available at http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/
• Performed task: Part-of-Speech tagging and lemmatization in Segmenter-POS-Tagger
• Input/Output formats: AlvisNLP/ML internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML, Linux/Mac-OS/Windows
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: Parameter files available for several languages. The English parameter file is used in this use case.

Genia Tagger
• Version: 0.5rc (relies on Genia Tagger version 3.0.2)
• License: ALv2 (for AlvisNLP/ML); Genia Tagger 3.0.2: “Redistribution and use in source and binary forms, with or without modification, are permitted for non-commercial purposes.”
• Link to documentation: http://www.nactem.ac.uk/GENIA/tagger/ (original doc)
  http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/GeniaTagger/doc (AlvisNLP/ML doc)
• Link to the code: http://www.nactem.ac.uk/tsujii/GENIA/tagger/geniatagger-3.0.2.tar.gz
• Performed task: Part-of-Speech tagging and lemmatization in Segmenter-POS-Tagger
• Input/Output formats: Alvis internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML, Linux, gcc compiler
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: POS-tag/chunk/EN models

\textsuperscript{18}http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/Tagger-Licence
8.1.3.4 Entity recognition and normalization tools

The AS-D NER and Normalizer diagram in the appendices shows the role of the Entity recognition and normalization tools.

**SimpleProjector2**
- Version: 0.5rc
- License: ALv2
- Performed task: Lexicon projection. Used in AS-D NER and Normalizer
- Input/Output formats: AlvisNLP/ML internal representation format
- Vocabulary format: tabular text file (one entry per line)
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
- Performance requirements and limits (used memory, processing speed, multi-threading): RAM for dictionary and corpus size
- Used static resources and models: may be used with any lexicon (in the required format)

**RegExp**
- Version: 0.5rc
- License: ALv2
- Performed task: Matches a regular expression on sections contents and create an annotation for each match. Used in AS-D NER and Normalizer
- Input/Output formats: AlvisNLP/ML internal representation
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
- Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
- Used static resources and models: Regular expressions

**PatternMatcher**
- Version: 0.5rc
- License: ALv2
- Link to documentation: [http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/PATTERNMatcher/doc](http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/PATTERNMatcher/doc)
- Performed task: Matches a specific Pattern (with previous annotation) on sections contents and create an annotation for each match. Used in AS-D NER and Normalizer
- Input/Output formats: Alvis internal representation
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: Regular expression-like patterns

YateaExtractor
• Version: 0.5rc (relies on BioYaTeA CPAN module version 0.11)
• License: ALv2 (for AlvisNLP/ML) and The Perl 5 License (Artistic 1 & GPL 1)\(^\text{19}\) (for BioYatea CPAN module)
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/YateaExtractor/doc
• Performed task: Term extraction (using the BioYaTeA tool) in AS-D NER and Normalize
• Input/Output formats: Alvis internal representation, extracted terms in XML file
• Deployment mode: AlvisNLP/ML module, CPAN module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML, CPAN
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: term extraction patterns

ToMapProjector
• Version: 0.5rc
• License: ALv2
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/TomapProjector/doc
• Performed task: Normalize entities according to an ontology or terminology (used in AS-D NER and Normalizer)
• Input/Output formats: Alvis internal representation, XML serialized lexicon analysis, YaTeA extracted terms as XML
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, and limits (used memory, processing speed, multi-threading) RAM for corpus and lexicon size
• Used static resources and models: term extraction, pre-analyzed lexicon/ontology

ToMapTrain
• Version: 0.5rc
• License: ALv2
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/TomapTrain/doc
• Performed task: Analyze the terms of an ontology/lexicon in order to use them with the normalization module (ToMapProjector). Used in Ontology/Lexicon Analyzer.
• Input/Output formats: Alvis internal representation, XML serialized lexicon analysis

\(^{19}\) http://dev.perl.org/licenses/
8.1.3.5 Tools for communication between TDM and URGI parent applications

8.1.4 Applications

The AS-D Data Exporter diagram shows conversion tools from the TDM application formats to the parent applications ones.

Annotation Export

- **Version:** Alvis
- **Performed task:** Transform annotations in OMTD format into annotations in the external OMTD-URGI format as a file. Used in AS-D Data Exporter.
- **Input formats:** Alvis/OMTD internal format
- **Output formats:** external OMTD-URGI format
- **Deployment mode:** workflow component

Wheat KB export

- **Performed task:** build an archive into the OMTD-URGI format of the updated versions of the KB maintained by MaLAGE. To be used by URGI in their search engine. Used in AS-D Data Exporter. The Wheat Trait Ontology is part of it.
- **Input formats:** OMTD internal format
- **Output formats:** external OMTD-URGI format
- **Deployment mode:** workflow component

Wheat KB import

- **Performed task:** convert the archive of the updated versions of the KB received from URGI into OMTD KB internal format. The lexica listed above: wheat taxon list and marker lexicon are parts of it. The merged gene lexica will be maintained at URGI and imported by OMTD.
- **Input formats:** external OMTD-URGI format
- **Output formats:** OMTD internal format
- **Deployment mode:** workflow component

8.1.5 Services

NCBI Taxonomy download

- **Performed Tasks:** retrieve the last version of the NCBI Taxonomy
- **Input formats:** NA
• Output formats: tar, gz, proprietary
• Terms of use (esp. performance limitations): https://uts.nlm.nih.gov/license.html free of use, not OA/OSS
• Hosting institution: NCBI
• Type of interface: FTP download

Elsevier API
• Address: http://dev.elsevier.com/
• Performed Tasks: get full text files from DOIs
• Input formats: API query with a list of DOIs
• Output formats: XML or HTML
• Terms of use (esp. performance limitations): (source: policy 20 for TDM use cases) “We have adopted a license–based approach which enables researchers at subscribing institutions to register for an API key to text mine for non-commercial research purposes and to gain access to full text content in XML for this purpose.” “In addition to supporting researchers at subscribing institutions to text mine, our policy also allows the following: Open access content: Everyone can download open access content to mine via our Full Text API, in line with the individual article’s user license.”
• Performance limitations: “there are no hard limits on the number of items that may be downloaded via our API. Nevertheless, a reasonable and customary rate limit remains in place to ensure equal access to the API for all users, and we continue to ask users to use our service responsibly.”
• Hosting institution: Elsevier
• Type of interface: RESTful

Springer API
• Address: https://dev.springer.com/
• Performed Tasks: get full text files from DOIs
• Input formats: API query with a list of DOIs
• Output formats: PDF
• Terms of use (esp. performance limitations): (source: policy 21) “Springer grants text- and data-mining rights to subscribed content to researchers via their institutions, provided the purpose is non-commercial research. “ “Publications or analyses resulting from TDM of subscribed content may include quotations from the original text of up to 200 characters, or 20 words, or 1 complete sentence. They should cite the original Springer content in the form of a DOI link. Permission to reproduce images may be granted on a case-by-case basis. For Open Access (OA) publications from Springer, BioMed Central and SpringerOpen, TDM is usually allowed without restrictions since the majority of our OA content is licensed under CC-BY.”
• Performance limitations: “TDM researchers are requested to be considerate and limit their downloading speed to a reasonable rate.”

---

20 https://www.elsevier.com/about/company-information/policies/text-and-data-mining
• Hosting institution: Springer
• Type of interface: RESTful

PMC OA Web Service
• Address: https://www.ncbi.nlm.nih.gov/pmc/utils/oa/oa.fcgi
• Performed Tasks: get bibliographic metadata and document URL full text files from DOIs
• Input formats: doi
• Output formats: PDF
• Terms of use (esp. performance limitations): (source: Developer resources) “The use of our APIs is entirely free, and doesn’t require an API key”. “Include two parameters that help to identify your service or application to our servers: tool should be the name of the application, as a string value with no internal spaces, and email should be the e-mail address of the maintainer of the tool, and should be a valid e-mail address.”
• Performance limitations: “Do not make concurrent requests, even at off-peak times”
• Hosting institution: NCBI
• Type of interface: REST

Digital Object Identifier System
• Address: https://www.doi.org
• Performed Task: get URL from DOIs
• Input formats: URL query: composed of https://doi.org/followed by DOI number. e.g. https://doi.org/10.1007/978-3-319-13674-5_28
• Output format: URL of the landing page
• Performance limitations: unknown
• Hosting institution: DOI Foundation
• Type of interface: http

CrossRef API
• Address: http://www.crossref.org
• Performed Task: get URL (resource field) and bibliographic metadata from DOIs
• Input formats: Open URL query or http query
• Output format: xsd_xml or UNIXREF format. See http://help.crossref.org/doi-to-metadata-query
• Terms of use (esp. performance limitations): (source: http://www.crossref.org/01company/free_services_agreement.html) “CrossRef asserts no claims of ownership to individual items of bibliographic metadata and associated Digital Object Identifiers (DOIs) acquired through the use of the CrossRef Free Services. Individual items of

---

22 https://www.ncbi.nlm.nih.gov/pmc/tools-developers/
23 http://help.crossref.org/doi-to-metadata-query
bibliographic metadata and associated DOIs may be cached and incorporated into the user's content and systems. "...

- Best practice: use XML
- Performance limitations: Restricted to registered users
- Hosting institution: Crossref
- Type of interface: REST

8.1.6 Other

Web of Knowledge
- Address: http://apps.webofknowledge.com/
- Performed Tasks: identify lists of relevant scientific publications (Corpus Builder)
- Input formats: request by human user

PubMed
- Address: https://www.ncbi.nlm.nih.gov/pubmed
- Performed Tasks: identify lists of relevant scientific publications (Corpus Builder)
- Input formats: request by human user

8.2 Deployment Plan

The application is initially deployed on Migale infrastructure. Migale is a certified bioinformatics service platform that provides web application hosting, RDBMS hosting, virtualization, storage, and a cluster with more than 1000 cores for heavy data processing.

The Deployment Plan of the AS-D application follows the same principles as the one of the AS-C application (Chapter 5).

The initial deployment will use INRA TDM software infrastructure, Alvis Suite, including:

- AlvisNLP/ML: corpus processing, document format conversion, and model training.
- AlvisAE: training corpus annotation.
- AlvisIR: semantic search engine.

This initial deployment will handle only the Application Design scenario and part of the Production Processing scenario. Indeed update scenarios will rely on features of the future OMTD infrastructure.

The redeployment on the OMTD infrastructure will take place gradually in several steps detailed below.

Registry entries

The most critical and reusable components and resources will have an entry in the OMTD registry.

Alvis components as OMTD components
Alvis software components will be wrapped as OMTD components so that they can be included in OMTD workflows. The integration of components will take place in several progressive steps.

AlvisNLP/ML is the first component to be integrated. It can serve as a generic corpus processing component. The integration of AlvisNLP/ML will allow to redeploy most of the automatic corpus processes described in this document on the OMTD infrastructure. Then individual components of AlvisNLP/ML may be deployed individually so that other workflows can take advantage of their functionality. The individually redeployed components will be selected according to their reusability by other partners and use-cases and originality.

The manpower necessary for wrapping AlvisNLP/ML and its components depends on:

- the API provided by OMTD for wrapping components,
- how much the OMTD data interface is compatible to AlvisNLP/ML data model
- the amount of OMTD-SHARE metadata input required.

As an annotation editor, AlvisAE will be integrated through the AERO protocol (OMTD internal document) which allows OMTD to import and export data from and to annotation editors, as well as to control annotation projects.

**Integration with the host application**

One of the greatest benefits anticipated from the OMTD infrastructure is the ability to build workflows that can communicate with external applications, the host application in this use-case. The most appealing features are:

- The management of different versions of resources and components; the ability to chose the version for each
- The querying of the environment of a workflow: which resources are used? Which elements have a newer version?
- The querying of metadata associated to different elements, in particular, for annotated corpora, the details of the processing environment (resource versions for instance).
- The programmatic control and monitor of workflow operations: the host application will need to be able to launch the Production Processing, monitor its advancement and download the result with as few human interventions as possible.
- Logging facilities and error handling: how easily the host application manager can monitor the progress of the corpus processing, be notified of errors, and take action about errors.

### 8.3 Data Interfaces

The AS-D application takes as input three types of knowledge bases (see format in section 6.1.2)

- External resources (=NCBI taxonomy),
- KB maintained by MaLAGE, the infrastructure that supports the TDM application (=WheatPhenotype ontology, Latin expression lexicon, Stopword lexicon)
• KB maintained by the URGI, the infrastructure that supports the parent application (=Gene and Marker lexica, Wheat taxon lexicon, WIPO Wheat Phenotype Ontology)

The two first sets of KB are exported by the TDM application to the parent application by the Wheat KB export tool (section 6.1.3.5).

The latter is imported by the AS-D TDM application from the parent application by the Wheat KB import tool (section 6.1.3.5).

The AS-D application also takes corpora as input (see format in section 6.1.1)

• the initial Wheat corpus and its updates are handled by the Corpus Builder
• the SAM Blé training corpus

The AS-D TDM application exports text annotations to the parent application. It is done by the Annotation Export tool (see format in section 6.1.3.5).

8.4 User Interfaces

The user interfaces in the parent applications that include TDM results are of two kinds.

GnplIS interface

GnplIS\(^{24}\) is a modular and interoperable information system dedicated to plant and pest genomics that enables scientists to mine genomic and genetic data. GnplIS is accessible through a public web portal allowing users to browse each data module or to search through all the data simultaneously using a quick search ('google like search') or advanced search (Biomart, Galaxy, Intermine) tools. It needs to be adapted to allow to display OMTD results.

WheatIS search

WheatIS search\(^{25}\) is a single entry point for the wheat community to find available data through a full text search engine, allowing searching the central repository and the databases of the platform network dynamically. It is fed with GnplIS data among others. It provides a quick search interface that allows the federated query of several relevant databases of the domain. It requires minimal adaptation to integrate OMTD results.

Screen mock-ups are shown in the appendices.

8.5 Data Processing Scenarios

The scenarios defined below refer to processes that are represented by diagrams in Annex 3.

There are three main scenarios that represent application life cycle phases:

\(^{24}\)https://urgi.versailles.inra.fr/Tools/GnplIS
\(^{25}\)https://urgi.versailles.inra.fr/wheatis/

TDM application production: production and export of annotations

TDM application update: curation, corpus and KB update, re-training.

(The detail below contains some overlapping material with Chapter 5 and 7 due to the similarity of scenarios, and that has been replicated deliberately for completeness of each part.)

**Application Design**

During the Application Design (AS-D Application Design diagram in Appendix 3), we gather the resources and software components necessary to the initial deployment of the application. At the end of this scenario, all the resources and components are ready for the first production processing of the corpus.

This is the scenario where we make the remaining operational choices with regards to the specifications. For instance if several software components were considered for a single step, then one of them is finally selected by operational criteria: availability, ease of installation. The same approach is used to handle resource alternatives. The TDM application manager in agreement with the host application manager will make these choices.

Choices to be made in this use-case:

- The ontology to be used to identify phenotypes;
- POS-tagger: TreeTagger or Genia Tagger

The choices will be motivated by the availability of the software or resource, how easily they can be wrapped into OMTD, and by their performance (e.g. F-score, computation time). Each step of the automatic annotation will be evaluated separately. Metric evaluation will be used if a reference corpus is available for a particular task. Otherwise, end-users and the application manager will evaluate the automatic annotation by sampling the results and assessing their accuracy.

Currently we plan to use the SAM Blé corpus.

One of the critical resources to build during the Application Design is the initial document set using the process described in the Corpus Builder diagram (Appendix 3). The result if the process is both the corpus that will be processed, and the query that will be used to keep the corpus up to date. The corpus design involves a close cooperation between the application manager, host application manager, and domain-expert end-users.

Finally, the Application Design includes the acquisition of resources specific to this application: reference lists for taxa, genes and markers.

**Production processing**
The Production Processing scenario (AS-D Production Processing diagram in Appendix 3) represents the normal lifecycle of the application. The processing of the corpus is triggered automatically and periodically. The resources, corpus, and components used are the most up-to-date at the time this scenario runs.

This scenario includes the automatic annotation of the corpus (AS-D Automatic Corpus Annotator diagram in the appendices), then the transmission of the annotation to the host application (AS-D Data Exporter diagram in the appendices).

The frequency of the Production processing scenario runs will depend on the processing time of the whole corpus and the expected turnover of the resources and the query results. This frequency may be adjusted during the lifecycle of the application. A timer (like cron) thus triggers the Production processing update component and the rest of the process is fully automatic.

It is expected that the Production process is incremental: it should only process new documents added to the corpus (see Resource Update scenario), unless one of the external resources is updated, in which case the whole corpus is re-processed.

Each time the update component triggers the Production processing scenario, it logs the specifics of the new version of the data. This log can be consulted and shall include the following information: a timestamp, the version of each component and resource, and the list of new references.

**Resource Update (training)**

The Resource Update scenario is the set of procedures that ensure that the most up-to-date resources are used during the Production Processing, and that both the TDM application and the host application refer to the same version of each resource. These procedures are automated as much as possible; however the distribution mode of some base material may require human supervision. The Resource Update scenario involves diverse processes:

**Corpus increment**

The query is re-run periodically in order to monitor new documents relevant to the application. Depending on the database used for gathering references, this step may be automated. If not the TDM application manager will run the query and fetch the results. Nevertheless, the following steps for building the corpus (Corpus Builder diagram in the appendices) are triggered automatically.

**Acquisition of new versions of external resources**

The following resources and software are provided by external entities: NCBI taxonomy, list of markers and genes provided by the parent application and the ontology of phenotypes. When these entities provide means to monitor updates, and to download automatically new versions, then this may be done automatically. However, some situations may require the validation of the application manager. For instance, major version shifts of resources may change the annotation results in such a way that
they become unfamiliar to the end-users. Additionally changes in software behaviour may break the interoperability of the corpus annotation process.

Some resources have to be pre-processed and transformed in order to be used by software components of the corpus annotation (Ontology/Lexicon Analyzer diagram in Appendix 3). These processes are automatically triggered upon the arrival (and the validation if necessary) of a new version of the base resource.

However, most of the resources are meant to be used for a single tool and they are often embedded in the tool distribution. In this case a new version of a resource might be exposed as new version of the tool itself. This is mostly the case for generic and linguistic resources (POS models, generic NE resources, etc.)

Finally, some resources are also used by the host application (AS-D Data Exporter diagram in the appendices). Thus the host application must be informed when a new version of a resource is used and how this version can be accessed. This communication is automated in the same way as resource pre-processing.

8.6 Limitations

Access to the documents

The licensing and technical restrictions with regards to the automatic processing of some documents necessary to the use case are not clear yet. The whole corpus necessary to obtain useful data for the end-user may not be available. For instance, the reference may not be available for direct download, or the harvester developed by Open University might be barred from retrieving them.

To overcome this limitation the strategy adopted is to develop first resources that would maximize the amount of documents in the corpus.

Depending on the needs, specific harvesters could be developed by Open University to complete the panel of tools for corpus collection.

GRNET Infrastructure

As mentioned in section 2, legal or technical restrictions might force us to deploy another instance of OMTD on URGI infrastructure. In this case the manpower necessary to the deployment will hinder the software integration of the use case into OMTD.

8.7 Release Plan

January 2017: V1

OMT deployment

- Entry of main components in the OMTD registry.
• Restrictions: availability of metadata for components.

Application development

The 1st release of the Wheat application has been developed using the Alvis platform. The objective is to demonstrate that the complete pipeline is operational from data providers to data integrators.

This 1st release is limited in terms of:

• Corpus scope: only *Triticum aestivum* has been considered in a subset of all possible sources (= content providers): large set of references from bibliographic database.
• Types of entities to be recognized: taxa, genes and markers
• Data integration in the final application will be restricted to database and to the WheatIS search application on a development server (no interface adaptation yet for GnpIS).
• Data interfaces: data transfers between the final application and the TDM application is performed manually.
• Entries to the OMTD registry: AlvisNLP/ML as a whole, resources.

August 2017

OMT deployment

• Integration of AlvisNLP/ML in OMTD workflow engine as a Docker image.
• Storage of main resources in the OMTD infrastructure.
• Deployment of the Wheat Data TDM application on OMTD using a Docker image.
• Scenarios: Production Processing periodic update.
  o Restriction: the GRNET OMTD instance can be used.

Application development

• Data integration in the client application at URGI: adaptation of the GUI, database and indexing of WheatIS / GnpIS on a development server.
• Data interfaces: data transfers between the final application and the TDM application through stable URL.
• Additional type of entities to be recognized and categorized using WTO: phenotypes

January 2018

OMTD deployment

• Entry of main components in the OMTD registry.
• Integration of the main individual AlvisNLP/ML components, and of resource processing tools (see section 5.1.3 Components/tools) using Docker images.
• Scenarios: Resource Update.
• Development of corpus design tools.
• Restriction: the GRNET OMTD instance can be used.

Application development

• Data will be fully integrated into the final application and all user interface adaptation will be achieved. It includes querying the database by using the phenotype ontology and synonyms.
• Interoperability: the building of the correspondence between the text phenotype ontology (WTO) and the experimental data phenotype ontology (WIPO) will be achieved in order to index all phenotypes with the same resource.
• In terms of corpus, the application may benefit from more sources provided by the OpenMinTeD platform or, if not available, by Open University harvesters.
• Additional type of entities to be recognized: wheat varieties.
• KB import and export will be automated whenever relevant.

8.8 References

Claire Nédellec, Robert Bossy, Dialekti Valsamou, Marion Ranoux, Wiktoria Golik, Pierre Sourdille. Information Extraction from Bibliography for Marker Assisted Selection in Wheat. In proceedings of Metadata and Semantics for Agriculture, Food & Environment (AgroSEM’14), special track of the 8th Metadata and Semantics Research Conference (MTSR’14), Springer Communications in Computer and Information Science, Series Volume 478, Karlsruhe, pp 301-313, Germany, 2014. DOI: 10.1007/978-3-319-13674-5_28
9. AS-E: Extracting gene regulation networks involved in seed development (SeeDev)

This chapter contains the specification of the design of the SeeDev application. The document provides the information that are required by the implementation, testing and maintenance of the application. Its purpose is also to anticipate the reuse and reconfiguration of the AS-E workflow by OMTD users by providing description of all necessary components including ancillary tools.

It has been written as a stand-alone document that can be read in isolation and used as such by application designers. It contains some overlapping material with AS-C and AS-D chapters that has been replicated deliberately for completeness.

The SeeDev TDM application is part of a parent application called FLAGdb++ at IPS2

NLP steps pre-processes documents, namely tokenization, sentence segmentation, lemmatization, POS tagging and syntactic parsing. These analyses will serve as resources for subsequent and more complex IE tasks.

Two types of entities are handled, entities defined in nomenclatures such as genes and entities defined in ontologies such as pathways. Entity recognition and categorization identify the entities in the documents and map them to the relevant knowledge sources (e.g. nomenclature, ontologies). The workflow uses a combination of NLP and ML methods trained on the SeeDev public reference corpus.

In a next step, relations between tagged entities are extracted, based on machine learning.

The TDM results will be used by the target bioinformatics FLAGdb++ application. Beyond extracted information, the TDM workflow shares three types of information with FLAGdb++: the knowledge base references (e.g. genes), the metadata for textual information display (e.g. snippet) and user feedback information. The first type of information, knowledge base references, requires synchronization of the updates between the TDM pipeline and the parent application since it comes from external resources.

Integration of TDM results in the bioinformatics FLAGdb++ application need a mapping between TDM entities and biological entities. Some of them have currently not equivalent in FLAGdb++ application, such as Tissue, Pathway or Development Phase. The SeeDev client application evolves, taking into account material and human constraints due to the integration of TDM results in the client (FLAGdb++) application.

26 http://urvg.evry.inra.fr/FLAGdb
27 http://www.ips2.u-psud.fr/?lang=en
9.1 Resources

9.1.1 Document Formats
The application uses two kinds of document corpora, (1) corpora for training the machine learning methods that will be used for IE prediction called SeeDev and (2) a production corpus from which the TDM pipeline extracts the relevant information, called PlantDev.

**SeeDev corpus**
The SeeDev corpus is a reference corpus with manual annotation of regulations involved in plant development. It is part of BioNLP Shared Task 2016. It is in BioNLP-ST format\(^{28}\).

- Link to specification/documentation: Representation and download (BioNLP-ST'16)\(^ {29}\)
- Intended Use: Training of ML for IE. Manual reference annotation has been achieved by plant science experts (see [Chaix et al., 2016] for details on the reference annotation).
- License: CC-BY-SA license v3.0. (INRA)

**Plant Development corpus (PlantDev)**
The PlantDev corpus contains all relevant abstracts and full-text papers for the Use Case that are technically and legally accessible. The formats of the documents depend on their source, which are diverse. They are detailed below.

- Link to specification/documentation: see appendices.
- Intended Use: Extraction of relevant information. Periodic update of the corpus with new publications that fit the criteria.
- License: when available, formal description of licenses are stored.

**XML Elsevier articles (part of PlantDev corpus)**

- Link to specification/documentation: Journal Article (JA) DTD 5.4.0\(^ {30}\)
- Intended Use: Extraction of relevant information. Periodic update of the corpus with new publications that fit the criteria.
- License: Can be OA or under subscription. Formal description of licenses are stored.
- Format: XML
- API: http://dev.elsevier.com/

**HTML Springer articles (part of PlantDev corpus)**

- Link to specification/documentation: How to\(^ {31}\) and policy
- Intended Use: Extraction of relevant information. Periodic update of the corpus with new publications that fit the criteria.

---

\(^{28}\) http://2013.bionlp-st.org/file-formats

\(^{29}\) http://2016.bionlp-st.org/tasks/seedev

\(^{30}\) https://www.elsevier.com/authors/author-schemas/elsevier-xml-dtds-and-transport-schemas

9.2 Community Driven Applications Design Report

- License: Can be OA (see article licenses\textsuperscript{32}) or under subscription. Formal description of licenses are stored.
- Format: HTML
- API: https://dev.springer.com/

PDF Springer articles (part of PlantDev corpus)
- Link to specification/documentation: How to and policy
- Intended Use: Extraction of relevant information. Periodic update of the corpus with new publications that fit the criteria.
- License: Can be OA or under subscription. Apparently no formal description of licenses
- Format: PDF
- API: https://dev.springer.com/

PDF PMC articles
- Intended Use: Extraction of relevant information. Periodic update of the corpus with new publications that fit the criteria.
- License: OA
- Format: PDF
- API: https://www.ncbi.nlm.nih.gov/pmc/utils/oa/oa.fcgi

Text PlantDev corpus
The Text PlantDev Corpus contains all relevant abstracts and full-text papers for the AS-E Use Case in text format. It is obtained by the conversion of the PlantDev corpus into text (see above).
- Intended Use: Extraction of relevant information. The corpus will be periodically updated with new publications.
- Format: OMTD
- License: same as PlantDev corpus.

9.1.2 Knowledge Bases
This section includes the list of all knowledge bases that are potentially used (vocabularies, ontologies, databases, lexicons etc.) for text annotation. Most of them are external resources. Their versions can therefore evolve. They will be maintained and processed (selection, completion, rewriting) by either FLAGdb++ team (section 7.1.2.1) or SeeDev TDM team (sections 7.1.2.2 and section 7.1.2.3). The assignment of the KB management between the teams may evolve along the time.

9.1.2.1 Resources from FLAGdb++ application
Part of the external resources of the TDM application are provided by the FLAGdb++ parent application in particular information about plant genomes for entity recognition and normalization. To date, the detail of all resources managed by FLAGdb++ can be found at http://tools.ips2.u-

\textsuperscript{32}https://www.elsevier.com/about/company-information/policies/open-access-licenses
psud.fr/projects/FLAGdb++/HTML/data.shtml (column “origin”). In those cases where other external resources are similar and / or overlapping with those provided by FLAGdb++, we focused priority on the resources provided by FLAGdb++.

**FLAGdb++ resources**

- **Version:** December 2016 release
- **Link to resource (url):** FLAGdb++ application could be downloaded from http://tools.ips2.usud.fr/projects/FLAGdb++/Appli/FLAGdb.jnlp, but the resources are not available online.
- **Link to documentation:** The 'DATA' section describes all the information managed by FLAGdb++ either from other infrastructures or from high-throughput experiments or bioinformatics prediction pipelines. The data sources, the graphical visualization, the associated tools and the integration methods are described. The data numbers for each feature is detailed in the Data chapter that mainly describes the data for Arabidopsis thaliana but the integration methods used for the other genomes are similar.
- **Intended use:** AS-E NER Preprocessor: Fixed entity annotation + AS-E Entity Recognizer + AS-E Entity Normalizer
- **Data format:** To be defined. It is possible that we need a tool to convert this data (see the tool “PlantDev KB import”).

### 9.1.2.2 Resources from TAIR

The Arabidopsis Information Resource (TAIR)\(^{33}\) maintains a database of genetic and molecular biology data\(^{34}\) for the model higher plant Arabidopsis thaliana\(^{35}\). Data available from TAIR includes the complete genome sequence along with gene structure, gene product information, gene expression, DNA and seed stocks, genome maps, genetic and physical markers, publications, and information about the Arabidopsis research community. Many resources are available on the TAIR platform (Description of all resources available at https://www.arabidopsis.org/download/index.jsp), including those described below to be used as vocabularies and lexicons. Some features are the same among the resources and described here:

- **Licence, terms of use:** TAIR term of use\(^{36}\): “Your use of the Service is limited to non-commercial, academic or not-for-profit research.” [...] “After April 1, 2014, you will require a TAIR account and a paid subscription to access some portions of the Service. Your account is for your personal use only and access to the Service via your account by others is not permitted.”

**Gene description (TAIR)**

- **Version and link:** http://www.arabidopsis.org/download_files/Genes/gene_aliases_20130831.txt
- **Link to documentation:**

\(^{33}\) https://www.arabidopsis.org/index.jsp

\(^{34}\) https://www.arabidopsis.org/about/datasources.jsp

\(^{35}\) https://www.arabidopsis.org/portals/education/aboutarabidopsis.jsp

\(^{36}\) https://www.arabidopsis.org/doc/about/tair_terms_of_use/417
• Araport and TAIR’s genome release files from Arabidopsis (Read me file)37
• Intended use: AS-E NER Preprocessor: Fixed entity annotation + AS-E Entity Normalizer (Gene, Protein resources)
• Data format: tabular text file
• Approximate size (MBs, number of entities/classes): 876 KB

Gene families of Arabidopsis thaliana (TAIR)
• Version: Gene families: gene_families_sep_29_09_update.txt
• Link to documentation: gene family data (GeneFamily_README.txt)
• Intended use: AS-E NER Preprocessor: Fixed entity annotation + AS-E Entity Normalizer (Gene families and Protein families)
• Data format: tabular text file
• Approximate size (MBs, number of entities/classes): 4,49 Mbs

9.1.2.3 Other external resources

MirBase
• Version: the current one, release 21 (2014) (to be confirmed)
• Link to resource (url): mirbase.org
• Link to documentation: miRBase: “the microRNA database” miRBase provides the following services: The miRBase database is a searchable database of published miRNA sequences and annotation. ftp://mirbase.org/pub/mirbase/CURRENT/README
• Intended use: AS-E NER Preprocessor: Fixed entity annotation + AS-E Entity Normalizer (RNA)
• Data format: text
• Approximate size (MBs, number of entities/classes): Release 21 of the database contains 28645 entries representing hairpin precursor miRNAs, expressing 35828 mature miRNA products, in 223 species.
• Licence, terms of use: in README document: “miRBase is in the public domain. It is not copyrighted. You may freely modify, redistribute, or use it for any purpose. See ftp://mirbase.org/pub/mirbase/CURRENT/LICENSE for details.”

Rfam
• Version: To be confirmed with FLAGdb++ application (the current one is 12.1, April 2016)
• Link to resource (url): Rfam browser38 with query “Arabidopsis thaliana”, or Rfam ftp
• Link to documentation: The Rfam database is a collection of RNA families, each represented by multiple sequence alignments, consensus secondary structures and covariance models (CMs)
• Intended use: AS-E NER Preprocessor: Fixed entity annotation + AS-E Entity Normalizer (RNA)
• Data format: text
• Approximate size (MBs, number of entities/classes): 2474 families, a query on “Arabidopsis thaliana” found 177 families

37 http://www.arabidopsis.org/download_files/Genes/TAIR10_genome_release/README_TAIR10.txt
38 http://rfam.xfam.org/search#tabview=tab3
• Licence, terms of use: “Rfam is freely available and in the public domain under the Creative Commons Zero licence. See ftp://ftp.ebi.ac.uk/pub/databases/Rfam/CURRENT/COPYING for more information.”

Other

• Agris Binding Sites List (no longer required)
• Arabidopsis Hormone Database 2.0 (no longer required)
• BioCyc Database Collection - Pathway (no longer required)
• KEGG PATHWAY (no longer required)
• Reactome (no longer required)
• PANTHER DB Pathways (no longer required)
• NCBI taxonomy (no longer required)

Internal resources

StopWords

• Version: V1
• Intended use: Word Exclusion in AS-E NER Preprocessor
• Data format: text
• Approximate size (MBs, number of entities/classes): 179 words
• Licence, terms of use: CC-BY 2.0

Bibliographical reference Regular Expression

• Version: V1
• Link to documentation: None: Pattern for bibliographical reference detection such as (ABD, et al. 2000)
• Intended use: Bibliographical reference detection in AS-E NER Preprocessor: Fixed entity annotation
• Licence, terms of use: ALv2

9.1.2.4 Ontologies (no longer required)
The ontologies are used to assign standard categories to the complex entity types, namely Pathway. The relevant parts will be also transferred to be used by the parent FLAGdb++ application for query interpretation, user navigation and display.

• Gene Ontology (no longer required)
• Plant Ontology (no longer required)
• GRO (Gene Regulation Ontology) (no longer required)
• BTO (BRENDA Tissue and Enzyme Source Ontology) (no longer required)
• PATO (Phenotypic Quality Ontology) (no longer required)
• Plant Trait Ontology (no longer required)
• Gramene Taxonomy ontology (no longer required)
9.1.3 Tools/Components
Some of the components are the same as the Microbial Biodiversity application (AS-C, Chapter 5) and Linking Wheat To Literature (AS-D, Chapter 6). Their descriptions have been duplicated for sake of completeness.

9.1.3.1 Corpus construction tools
The diagram Corpus Builder (in the appendices) shows the role of the corpus construction tools.

**Extract DOIs**
- Performed task: read scientific literature records and extracts information on DOIs from them, removes DOI already present in the corpus
- Version: To Be Developed
- Input format: bibliographic database (WoS or PubMed)
- Output formats: XML

**Analyse publisher landing page**
- Performed task: check if the landing web page is correct, if not it searches for the URL of the full-paper. It outputs the number of successes.
- Version: To Be Developed
- Input format: web page
- Output formats: URL

The diagram Corpus Converter (see appendices) shows the role of the corpus converter tools.

**XML Reader (XMLReader2)**
- Performed task: Converts documents from publishers in XML/HTML format (Corpus Converter).
- Version: 0.5rc
- License: ALv2
- Input format: XML to OMTD internal representation
- Deployment mode: AlvisNLP/ML Module
- Requirements for system environment: AlvisNLP/ML
- Performance requirements and limits: RAM for corpus size
- Used static resources and models: the structure of the input XML is handled through the xlsTransform XSLT stylesheet.

**TextFileReader**
• Performed task: Converts documents from publishers in text format (Corpus Converter).
• Version: 0.5rc
• License: ALv2
• Input/Output formats: unicode text to AlvisNLP/ML internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters), AlvisNLP/ML
• Performance requirements and limits (used memory, processing speed, multi-threading): NA
• Used static resources and models: No

PDF Reader (TikaReader)
• Performed task: Converts documents from publishers in PDF format (Corpus Converter).
• Version: 0.5rc (requires Tika 1.13)
• License: ALv2
• Input/Output formats: PDF to AlvisNLP/ML internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters), AlvisNLP/ML
• Performance requirements and limits (used memory, processing speed, multi-threading): NA
• Used static resources and models: No

9.1.3.2 Linguistic pre-processing tools

WoSMig
• Version: 0.5rc
• License: ALv2
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/WoSMig/doc
• Performed task: Word segmentation (Tokenization) in AS-E NER
• Input/Output formats: AlvisNLP/ML internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: none

SeSMig
• Version: 0.5rc
• License: ALv2
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/SeSMig/doc
• Performed task: Sentence segmentation in AS-E NER
• Input/Output formats: Alvis internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: none

**TreeTagger**
• Version: 0.5rc (relies on tree-tagger version 3.2)
• License: ALv2 (for AlvisNLP/ML). TreeTagger 3.2: “freely available for research, education and evaluation. Usage of the system for commercial purposes is forbidden.” (TreeTagger License terms: http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/Tagger-Licence)
• Link to documentation: http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/ (original doc)
• http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/TreeTagger/doc (AlvisNLP/ML doc)
• Link to the code: Executable code (not open source) available at http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/
• Performed task: Part-of-Speech tagging and lemmatization in AS-E NER
• Input/Output formats: AlvisNLP/ML internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML, Linux/Mac-OS/Windows
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: Parameter files available for several languages; The English parameter file is used in this use case.

**Genia Tagger**
• Version: 0.5rc (relies on Genia Tagger version 3.0.2)
• License: ALv2 (for AlvisNLP/ML); Genia Tagger 3.0.2: “Redistribution and use in source and binary forms, with or without modification, are permitted for non-commercial purposes.”
• Link to documentation: http://www.nactem.ac.uk/GENIA/tagger/ (original doc)
• http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/GeniaTagger/doc (AlvisNLP/ML doc)
• Link to the code: http://www.nactem.ac.uk/tsujii/GENIA/tagger/geniatagger-3.0.2.tar.gz
• Performed task: Part-of-Speech tagging and lemmatization in AS-E NER
• Input/Output formats: Alvis internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML, Linux, gcc compiler
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: POS-tag/chunk/EN models

9.1.3.3 Entity recognition tools

**StanfordNER**
• Version: 0.5rc (uses StanfordNER 3.4)
License: ALv2 (for AlvisNLP/ML) and GNU General Public License (v2 or later) (for StanfordNER)

Link to documentation: Named Entity Recognition (NER) labels sequences of words in a text which are the names of things, such as person and company names, or gene and protein names. It comes with well-engineered feature extractors for Named Entity Recognition, and many options for defining feature extractors. http://nlp.stanford.edu/software/CRF-NER.shtml (original doc)

http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/StanfordNER/doc (AlvisNLP/ML doc)

Performed task: Named Entity Recognizer in AS-E NER Preprocessor: Fixed entity annotation

Input/Output formats: AlvisNLP/ML internal representation

Deployment mode: AlvisNLP/ML module

Requirements for system environment (OS, installed libs, language interpreters) AlvisNLP/ML and StanfordNER

Performance requirements and limits (used memory, processing speed, multi-threading) NA

Used static resources and models: NER model

**SimpleProjector2**

- Version: 0.5rc
- License: ALv2
- Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/SimpleProjector2/doc

Performed task: Lexicon projection in AS-E NER Preprocessor: Fixed entity annotation + AS-E Entity Recognizer + AS-E Protein and Gene disambiguation

Input/Output formats:

- AlvisNLP/ML internal representation format
- Vocabulary format: tabular text file (one entry per line)

Deployment mode: AlvisNLP/ML module

Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML

Performance requirements and limits (used memory, processing speed, multi-threading) RAM for dictionary and corpus size

Used static resources and models: may be used with any lexicon (in the required format)

**RegExp**

- Version: 0.5rc
- License: ALv2
- Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/RegExp/doc

Performed task: Matches a regular expression on sections contents and create an annotation for each match. Used in AS-E NER Preprocessor for fixed entity annotation

Input/Output formats: AlvisNLP/ML internal representation

Deployment mode: AlvisNLP/ML module

Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: Regular expression

PatternMatcher
• Version: 0.5rc
• License: ALv2
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/PatternMatcher/doc
• Performed task: Matches a specific Pattern (with previous annotation) on sections contents and create an annotation for each match. Used in AS-E Entity Recognizer
• Input/Output formats: Alvis internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: Regular expression-like patterns

WapitiTrain
• Version: 0.5rc (relies on Wapiti\textsuperscript{39} version 1.5.0)
• License: ALv2 (for AlvisNLP/ML); two-clause BSD License (for Wapiti, License terms: https://wapiti.limsi.fr/#licence)
• Link to documentation: https://wapiti.limsi.fr/manual.html (original doc)
• http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/WapitiTrain/doc (AlvisNLP/ML doc)
• Link to the code: https://github.com/Jekub/Wapiti
• Performed task: Training of a CRF model (to perform AS-E Entity Recognizer)
• Input/Output formats: AlvisNLP/ML internal representation (Wapiti toolkit integrated in AlvisNLP/ML)
• Deployment mode: AlvisNLP/ML module (executable)
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML, POSIX C compiler.
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: An annotated training corpus (here, the SeeDev corpus) and a configuration file to specify which features to use for training

WapitiLabel
• Version: 0.5rc (relies on Wapiti version 1.5.0)
• License: ALv2 (for AlvisNLP/ML); two-clause BSD License (for Wapiti, License terms: https://wapiti.limsi.fr/#licence)

\textsuperscript{39} https://wapiti.limsi.fr/
9.1.3.4 Entity normalization tools

YateaExtractor
- Version: 0.5rc (relies on BioYaTeA CPAN module version 0.11)
- License: ALv2 (for AlvisNLP/ML) and The Perl 5 License (Artistic 1 & GPL 1) (for BioYatea CPAN module)
- Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/YateaExtractor/doc
- Performed task: Term extraction (using the BioYaTeA tool) in AS-E Entity Normalizer
- Input/Output formats: Alvis internal representation, extracted terms in XML file
- Deployment mode: AlvisNLP/ML module, CPAN module
- Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML, CPAN
- Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
- Used static resources and models: term extraction patterns

ToMapProjector
- Version: 0.5rc
- License: ALv2
- Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/ToMapProjector/doc
- Performed task: Normalize entities according to an ontology or terminology (used in AS-E Entity Normalizer)
- Input/Output formats: Alvis internal representation, XML serialized lexicon analysis, YaTeA extracted terms (as XML file)
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, and limits (used memory, processing speed, multi-threading): RAM for corpus and lexicon size
• Used static resources and models: term extraction XML file, pre-analyzed lexicon/ontology

ToMapTrain
• Version: 0.5rc
• License: ALv2
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/TomapTrain/doc
• Performed task: Analyze the terms of an ontology/lexicon in order to use them with the normalization module (ToMapProjector). Used in Ontology/Lexicon Analyzer
• Input/Output formats: Alvis internal representation, XML serialized lexicon analysis
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, and limits (used memory, processing speed, multi-threading): RAM for corpus and lexicon size
• Used static resources and models: ontology/lexicon to be analyzed

9.1.3.5 Training and relation extraction tools

CCG parser
• Version: 0.5rc (relies on CCG Parser 1.00)
• License: ALv2
• Link to documentation: http://bibliome.jouy.inra.fr/demo/AlvisNLP/api/modules/CCGParser/doc
• Performed task: Dependency parsing in Relation extractor.
• Input/Output formats: AlvisNLP/ML internal representation
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for corpus size
• Used static resources and models: language models

TEES trainer
• Version: 2.0.1
• License: GNU General Public License
• Link to documentation: https://github.com/jbjorne/TEES/wiki/Training
• Link to the code: https://github.com/jbjorne/TEES
• Performed task: Train a corpus and produce a SVM-based learning model for TEES classifier (Relation extractor)
• Input/Output formats:
  o Interaction XML
  o BioNLP ST format
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters)
  o System: Unix/Linux
  o Language: Python
  o Dependencies: Python libs
• Performance requirements and limits (used memory, processing speed, multi-threading): RAM for training data set (esp. negative examples)
• Used static resources and models: Training and test corpora, feature definitions

**TEES classifier**

• Version: 2.0.1
• License: GNU General Public License
• Link to documentation: https://github.com/jbjorne/TEES/wiki/Training
• Link to the code: https://github.com/jbjorne/TEES
• Performed task: Predict (classify) relations from a corpus and SVM-based learning model (Relation extractor) Need Relation Extraction Model Learner
• Input/Output formats:
  o Interaction XML
  o BioNLP ST format
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters)
  o System: Unix/Linux
  o Language: Python
  o Dependencies: Python libs
• Used static resources and models: SVM-based Learning Model

**BioNLP format converter**

• Version: 2.0.1
• License: GNU General Public License
• Link to documentation: https://github.com/jbjorne/TEES/wiki/The-Preprocessor
• Link to the code: https://github.com/jbjorne/TEES/tree/master/Utils
• Performed task: Conversion between BioNLP ST format and Interaction XML in Relation extractor
  o Input/Output formats:
    o BioNLP ST format
• Interaction XML
• Deployment mode: AlvisNLP/ML module
• Requirements for system environment (OS, installed libs, language interpreters)
  o System: Unix/Linux
  o Language: Python
  o Dependency resolutions: Python libs
• Used static resources and models: None

**9.1.3.6 Tools for communication between TDM and FLAGdb++ applications**

**Annotation Export**

• Version: 0.5rc
• License: ALv2
- Performed task: Transform annotations in OMTD format into annotations in the external OMTD-FLAGdb++ format as a file in AS-E Production Processing
- Input/Output formats: AlvisNLP/ML internal representation, FLAGdb++ format
- Deployment mode: AlvisNLP/ML module
- Requirements for system environment (OS, installed libs, language interpreters): AlvisNLP/ML

**PlantDev KB export**

- Version: to be developed
- Performed task: build an archive into the OMTD-FLAGdb++ format of the updated versions of the BC maintained by MaIAGE. To be used by FLAGdb++. (AS-E Production Processing)
- Input/Output formats: OMTD internal format / external OMTD-FLAGdb++ format
- Deployment mode: Workflow component

**PlantDev KB import**

- Version: to be developed
- Performed task: convert the archive of the updated versions of the KB received from FLAGdb++ into OMTD BC internal format. Use for all resources provided by FLAGdb++. (AS-E Production Processing, AS-E NER Preprocessor: Fixed entity annotation + AS-E Entity Recognizer + AS-E Entity Normalizer)
- Input/Output formats: external OMTD-FLAGdb++ format / OMTD internal format
- Deployment mode: Workflow component

### 9.1.4 Services

**NCBI Taxonomy download**

- Performed Tasks: retrieve the last version of the NCBI Taxonomy
- Input formats: NA
- Output formats: tar, gz, proprietary
- Terms of use (esp. performance limitations): free of use, not OA/OSS
- Hosting institution: NCBI
- Type of interface: FTP download

**Elsevier API**

- Address: http://dev.elsevier.com/
- Performed Tasks: get full text files from DOIs (Corpus Builder)
- Input formats: API query with a list of DOIs
- Output formats: XML or HTML
- Terms of use (esp. performance limitations): (source: policy 41 for TDM use cases) “We have adopted a license–based approach which enables researchers at subscribing institutions to

register for an API key to text mine for non-commercial research purposes and to gain access to full text content in XML for this purpose.” “In addition to supporting researchers at subscribing institutions to text mine, our policy also allows the following: Open access content: Everyone can download open access content to mine via our Full Text API, in line with the individual article's user license.”

- Performance limitations: “there are no hard limits on the number of items that may be downloaded via our API. Nevertheless, a reasonable and customary rate limit remains in place to ensure equal access to the API for all users, and we continue to ask users to use our service responsibly.”
- Hosting institution: Elsevier
- Type of interface: RESTful

**Springer API**
- Address: https://dev.springer.com/
- Performed Tasks: get full text files from DOIs (Corpus Builder)
- Input formats: API query with a list of DOIs
- Output formats: PDF
- Terms of use (esp. performance limitations): (source: policy42) “Springer grants text- and data-mining rights to subscribed content to researchers via their institutions, provided the purpose is non-commercial research. “Publications or analyses resulting from TDM of subscribed content may include quotations from the original text of up to 200 characters, or 20 words, or 1 complete sentence. They should cite the original Springer content in the form of a DOI link. Permission to reproduce images may be granted on a case-by-case basis. For Open Access (OA) publications from Springer, BioMed Central and SpringerOpen, TDM is usually allowed without restrictions since the majority of our OA content is licensed under CC-BY.”
  Performance limitations: “TDM researchers are requested to be considerate and limit their downloading speed to a reasonable rate.”
- Hosting institution: Springer
- Type of interface: RESTful

**PMC OA Web Service**
- Address: https://www.ncbi.nlm.nih.gov/pmc/utils/oa/oa.cgi
- Performed Tasks: get full text files from DOIs (Corpus Builder)
- Output formats: PDF
- Terms of use (esp. performance limitations): (source: Developer resources43) “The use of our APIs is entirely free, and doesn't require an API key”. “Include two parameters that help to identify your service or application to our servers: tool should be the name of the application, as a string value with no internal spaces, and

---

41 https://www.elsevier.com/about/company-information/policies/text-and-data-mining
email should be the e-mail address of the maintainer of the tool, and should be a valid e-mail address.”

- Performance limitations: “Do not make concurrent requests, even at off-peak times”
- Hosting institution: NCBI

Digital Object Identifier System
- Address: https://www.doi.org
- Performed Task: get URL from DOIs (Corpus Builder)
- Input formats: URL query: composed of https://doi.org/ followed by DOI number. e.g. https://doi.org/10.1007/978-3-319-13674-5_28
- Output format: URL of the landing page
- Hosting institution: DOI Foundation

Crossref
- Address: http://www.crossref.org
- Performed Task: get URL (resource field) and bibliographic metadata from DOIs (Corpus Builder)
- Input formats: http://help.crossref.org/doi-to-metadata-query Open URL query or http query
- Output format: xsd_xml or UNIXREF format. See http://help.crossref.org/doi-to-metadata-query
- Terms of use (esp. performance limitations): (source: http://www.crossref.org/01company/free_services_agreement.html) “CrossRef asserts no claims of ownership to individual items of bibliographic metadata and associated Digital Object Identifiers (DOIs) acquired through the use of the CrossRef Free Services. Individual items of bibliographic metadata and associated DOIs may be cached and incorporated into the user’s content and systems. [..]”
- Best practice: use XML
- Performance limitations: Restricted to registered users
- Hosting institution: Crossref
- Type of interface: REST

9.1.5 Other
Web of Knowledge
- Address: http://apps.webofknowledge.com/
- Performed Tasks: identify lists of relevant scientific publications (Corpus Builder)
- Input formats: request by human user

PubMed
- Address: https://www.ncbi.nlm.nih.gov/pubmed
- Performed Tasks: identify lists of relevant scientific publications (Corpus Builder)
- Input formats: request by human user

9.2 Deployment Plan
The application is deployed on the IPS2 infrastructure in the parent application FLAGdb++ developed by Genomics Networks team. The Genomic Networks group develops different Bioinformatics and Biostatistics approaches to infer the biological function of *A. thaliana* genes and transcriptional regulatory elements. This team have developed mainly 2 databases (FLAGdb++, CATdb) that supports the IPS2 Transcriptomics platform and projects.

The Deployment Plan of the AS-E application follows the same principles as the one of the AS-C application (Chapter 5).

The initial deployment has used INRA TDM software infrastructure, Alvis Suite, including:

- **AlvisNLP/ML**: corpus processing, document format conversion, and model training.
- **AlvisAE**: prediction visualisation and curation.
- **AlvisIR**: semantic search engine.

This initial deployment handles the Application Design and part of the Production Processing scenario. Indeed all update scenarios will rely exclusively on features of the future OMTD infrastructure.

The redeployment on the OMTD infrastructure will take place gradually in several steps detailed here:

**Registry entries**

The most critical and reusable components and resources will have an entry in the OMTD registry.

**Alvis components as OMTD components**

Alvis software components will be wrapped as OMTD components so that they can be included in OMTD workflows. The integration of components will take place in several progressive steps.

AlvisNLP/ML is the first component to be integrated. It can serve as a generic corpus processing component. The integration of AlvisNLP/ML will allow to redeploy most of the automatic corpus processes described in this document on the OMTD infrastructure. Then individual components of AlvisNLP/ML may be deployed individually so that other workflows can take advantage of their functionality. The individually redeployed components will be selected according to their reusability by other partners and use-cases and originality.

The manpower necessary for wrapping AlvisNLP/ML and its components depends on:

- the API provided by OMTD for wrapping components, how much the OMTD data interface is compatible to AlvisNLP/ML data model,
- the amount of OMTD-SHARE metadata input is required.

---

As an annotation editor, AlvisAE will be integrated through the AERO protocol (OMTD internal working document)\textsuperscript{46} which allows OMTD to import and export data from and to annotation editors, as well as to control annotation projects.

**Integration with the host application**

One of the greatest benefits anticipated from the OMTD infrastructure is the ability to build workflows that can communicate with external applications, the host application in this use-case. The most appealing features are:

- The management of different versions of resources and components; the ability to chose the version for each.
- The querying of the environment of a workflow: which resources are used? Which elements have a newer version?
- The querying of metadata associated to different elements, in particular, for annotated corpora, the details of the processing environment (resource versions for instance).
- The programmatic control and monitor of workflow operations: the host application will need to be able to launch the Production Processing, monitor its advancement and download the result with as few human interventions as possible.
- Logging facilities and error handling: how easily the host application manager can monitor the progress of the corpus processing, be notified of errors, and take action about errors.

The local deployment of an OMTD instance on the FLAGdb++ infrastructure will not be considered due to external manpower constraints at IPS2.

### 9.3 Data Interfaces

The AS-E application takes as input three types of knowledge bases (see format in section 7.1.2)

- KB maintained by MalAGE, the infrastructure that supports the TDM application (= See Section Internal resources)
- KB maintained by IPS2, the infrastructure that supports the parent application (= See Section Resources from FLAGdb++ application)
- External resources (= all other resources listed in section Knowledge Bases)

The two first sets of KB are exported by the AS-E TDM application to the parent application by the KB export tool.

The latter is imported by the AS-E TDM application from the parent application by the KB import tool.

\textsuperscript{46} https://docs.google.com/document/d/1E75Ge2_n4EdCY7B9NKx8r00WsjVH4IlhK3jWnJEap4w/edit#heading=h.3wzpw5veff02
The AS-E application also takes corpora as input

- the initial PlantDev corpus and its updates are handled by Corpus Builder
- the SeeDev training corpus

The AS-E TDM application exports text annotations to the parent application. It is done by the Annotation Export tool.

9.4 User Interfaces

Data access for end-users (included in FLAGdb++)

FLAGdb++ is a relational database associated to a graphical interface written in Java (1.6). The rich client interface allows extended interactivity with the application and large graphical possibilities including the integration of heterogeneous data in FLAGdb++. FLAGdb++ application gives access to different plant genomes: the *Arabidopsis thaliana* genome is displayed by default. The main screens of FLAGdb++ can be found in appendix 2 "User Interfaces". The FLAGdb++ interfaces are described in more detail in FlagDB++ website\(^{47}\).

1. The data obtained by Information Extraction will be accessed in two different ways: keyword search (see Appendix 2: AS-E, Data access from a simple keyword search)
2. selection of the relevant gene from the chromosome representation to display the relations of this gene with others entities (see Appendix: AS-E, Data access from gene locus and relation search).

AlvisAE documentation includes the GUI for the project management and the end-user interactions. The main screens can be found in AlvisAE UI User Guide\(^{48}\).

9.5 Data Processing Scenarios

The scenarios defined below refer to processes that are represented by diagrams in Appendix 3.

There are three main scenarios that represent application life cycle phases:

- TDM application production: production and export of annotations
- TDM application update: corpus and KB update.

(The detail below contains some overlapping material with Chapter 5 and 6 due to the similarity of scenarios, and that has been replicated deliberately for completeness of each part.)

\(^{47}\) http://tools.ips2.u-psud.fr/projects/FLAGdb++/HTML/screenshots.shtml#Tut-index
Application Design

During the Application Design, OMTD supports automatic resource building (harvest, selection, and rewriting) and software component adaptation (mostly based on machine learning) that are necessary to the initial deployment of the application. At the end of this scenario, all the resources and components are ready for the first Production Processing of the corpus (see below).

This is the scenario where we make the remaining operational choices with regards to the specifications. For instance if several software components were considered for a single step, then one of them is finally selected by operational criteria: availability, ease of installation. The same approach is used to handle resource alternatives. The application manager in agreement with the host application manager will make these choices.

In this use-case, the most critical choices are:

- **POS-tagging**: TreeTagger or Genia tagger
- **NER**: Lexicon and rule-based approach vs. machine-learning (CRF) based approach (possibly a combination of the two)

The choices will be motivated by the availability of the software or resource, how easily they can be wrapped into OMTD, and by their performances (e.g. F-score, computation time). Each step of the automatic annotation will be evaluated separately. Metric evaluation will be used if a reference corpus is available for a particular task. Otherwise, end-users and the application manager will evaluate the automatic annotation by sampling the results and assessing their accuracy.

Currently we plan to use the following reference corpora:

- BioNLP-ST 2016 SeeDev corpus for NER and Normalization and Relation Extraction.
- Plant development corpus for NER and Normalization

One of the critical resources to build during the Application Design is the initial document set using the process described in Corpus Builder. The result of the process is both the corpus that will be processed with its metadata, and the query that will be used to keep the corpus up to date. The corpus design involves a close cooperation between the application manager, the host application manager, and domain-expert end-users.

Production processing

The Production Processing scenario represents the main lifecycle phase of the application. The processing of the corpus is triggered automatically and periodically. The resources, corpus, and components used are the most up-to-date at the time this scenario runs.
The Production scenario includes the automatic annotation of the corpus (AS-E Automatic Corpus Annotator), then the transmission of the annotation to the host application (See Data Exporter in Production Processing).

The frequency of the Production processing scenario runs will depend on the duration of the processing of the whole corpus and the expected turnover of the resources i.e. KBs and the query result paper collection. This frequency may be adjusted during the lifecycle of the application. The Production processing update component is thus triggered by a timer like cron and the rest of the process is fully automatic.

It is expected that the Production process is incremental: it should only process new documents added to the corpus (see Resource Update scenario), unless one of the external resources are updated, in which case the whole corpus is re-processed.

Each time the update component triggers the Production processing scenario, it logs the specifics of the new version of the data. This log can be consulted and shall include the following information: a timestamp, the version of each component and resource, and the list of new references.

**Resource Update**

The Resource Update scenario is the set of procedures that ensure that the Production Processing uses the most up-to-date resources, and that both the TDM application and the host application refer to the same version of each resource. These procedures are automated as much as possible, however the distribution mode of some base material may require human overseeing. The Resource Update scenario involves diverse processes:

**Corpus increment**

The bibliographic query is re-run periodically in order to monitor new documents relevant to the application. Depending on the bibliographic database used for gathering references, this step may be automated. If not the TDM application manager will run the query and fetch the results. Nevertheless, the following steps for building the corpus (Corpus Builder) are triggered automatically.

**Acquisition of new versions of external resources**

Resources provided by external entities are incorporated to the application by the PlantDev KB export . When these entities provide means to monitor updates, and to download automatically new versions, then this may be done automatically. However, some situations may require the validation of the application manager. For instance, major version shifts of resources may change the annotation results in such a way that they become unfamiliar to the end-users.

Some resources have to be pre-processed and transformed in order to be used by software components of the corpus annotation (Ontology/Lexicon Analyzer, CRF Model Learner, Relation...
Extraction Model Learner, Lexicon Lemmatization). These processes are automatically triggered upon the arrival (and the validation if necessary) of a new version of the base resource.

However, most of the resources are meant to be used for a single tool and they are often embedded to the tool distribution. In this case a new version of a resource might be exposed as a new version of the tool itself. This is mostly the case for generic and linguistic resources (POS models, generic NE resources, etc.)

Finally, some resources are also used by the host application provided by PlantDev KB export. Thus the host application must be informed when a new version of a resource is used and how this version can be accessed. This communication is automated in the same way as resource pre-processing.

9.6 Limitations

Access to the documents:

The licensing and technical restrictions with regards to the automatic processing of some documents necessary to the use case are not clear yet. The whole corpus necessary to obtain useful data for the end-user may not be available. For instance, the reference may not be available for direct download, or the harvester developed by OU might be barred from retrieving them.

To overcome this limitation the strategy adopted is to develop first resources that will maximize the amount of documents in the corpus.

Text quality:

The quality of the text for the linguistic tools must be the best to avoid predictions of lower quality. Converting text from HTML and PDF formats can be critical points for TDM pipeline operation.

9.7 Release Plan

9.7.1 1st release (January 2017)

OMTD deployment

- Entry of main components in the OMTD registry.
  - Restrictions: availability of metadata for components.

Application development

- Scenarios: Application Design, Production Processing (initial processing)
- Corpus subset: Abstracts about “Seed development of Arabidopsis” from PubMed (5089) and Web Of Knowledge (2000) with the query “(Arabidopsis AND Seed AND gene* AND (development OR maturation OR metabolism)”
  - Restrictions: harvester development and document harvest failure.
- Training corpus subset.
9.7.2 2nd release (August 2017)

OMTD deployment

- Integration of AlvisNLP/ML in OMTD workflow engine as a Docker image.
- Storage of main resources in the OMTD infrastructure.
- Deployment of the SeeDev Data TDM application on OMTD using a Docker image.
- Scenarios: Production Processing periodic update.
  - Restriction: the GRNET OMTD instance can be used.

Application development

- Integration of manual annotations and TDM predictions in INRA FLAGdb++
  - Restrictions: the integration of the TDM results requires a mapping of FlagDB++ entity types to FlagDB++ in order to align text annotation and FLAGdb++ resources. Lack of mapping with other entities than gene, protein, RNA and gene/protein family will prevent the integration of these TDM entity predictions.
- Prediction of relations between entities (i.e. Gene and Protein).
  - Restriction: quality of relation predictions.

9.7.3 3rd release (January 2018)

OMTD deployment

- Entry of main components in the OMTD registry.
- Integration of the main individual AlvisNLP/ML components, and of resource processing tools (see section 5.1.3 Components/tools) using Docker images.
- Scenarios: Resource Update.
- Development of corpus design tools.
  - Restriction: the GRNET OMTD instance can be used.

Application development

- Query and display of gene, protein, RNA and gene/protein family by FLAGdb++
- Development of corpus construction tools
- Abstract and full paper corpus (query extended to the whole plant development of Arabidopsis)
  - Restriction: availability to the harvesters.
- Improvement of quality of NER extraction (Gene, Protein and related types).
• Improvement of quality of prediction of relations between entities
  o Restriction: integration of relation extraction in OMTD depending on the quality of predictions.
• Evaluation tools (SeeDev task of BioNLP-ST).

9.8 References


10. **SS-A: Facilitation of complex information linking and retrieval from social sciences publications**

This application is going to do three things: 1) named entity recognition and linking, 2) variable mention detection and linking, and 3) keyword assignment. All those parts will in the end be integrated into the GESIS integrated information infrastructure, where several information objects (e.g. persons, publications, research data, projects) are present and partly interlinked. Our application will help to extend this linking even more.

The recognized named entities will be fed into our search index for literature retrieval, so that the search system in the digital library can assist the user searching for documents. Thus in ambiguous cases, when a query term can be both an entity and a regular noun, the system knows of this distinction and may provide the user with guidance like dividing the result set according to the type. Another use is a visual highlighting of entities in a text and the provision of links to e.g. Wikipedia where applicable.

The recognized variable mentions will be fed into our link database, where we have already established links between documents and datasets as Linked Open Data. The interlinked information can be presented to a user for more effective navigation, and may also be used for meta-analysis.

Automatically assigned documents’ keywords will be presented to a human curator (i.e. librarian) to facilitate his work. The curator will then add the correct keywords to article metadata.

### 10.1 Resources

#### 10.1.1 Document Formats

**Original article format**

- Intended use: text mining
  - this is the original format of documents that are input to the system
- Licenses of documents (only documents having a CC-BY licence can be shared):
  - Creative Commons - Attribution
  - Creative Commons - Attribution - ShareAlike
  - Creative Commons - Attribution - NonCommercial
  - Creative Commons - Attribution - NonCommercial - ShareAlike

**Free Digital Peer Publishing License**

**Article metadata**

- Intended use:
  - description of the PDF articles (title, authors, abstracts, language, licence, topic classification, keywords, etc.)
  - is used for filtering documents to create subcorpora
• If applicable: license, version, serialisation etc:
  o Serialisation: text/XML
  o License: CC0

**Machine-readable document format**

• Link to specification/documentation: UIMA XMI http://www.omg.org/spec/XMI/2.5.1/ ("XML (XML Metadata Interchange) is an OMG standard for the XML representation of object graphs")
• Intended use: text mining, standard machine-readable format for storage and retrieval, annotation
• If applicable: license, version, serialisation etc:
  o Version: depends on DKPro-Core version used
  o Serialisation: text/XML

**10.1.2 Knowledge Bases**

**Internal Resources**

**TheSoz (The Social Science Thesaurus)**

• Version: 0.93 (iQvoc version 4.9.0 (ges 16-0601))
• Link to documentation:
  o http://www.gesis.org/en/services/research/tools-zur-recherche/social-science-thesaurus/
• Intended use: document tagging (keyword assignment)
• Data format: SKOS-XL with proprietary extensions, Linked Data HTML representation, downloadable dump in RDF/Turtle
• Approximate size: about 12,000 entries (8,000 descriptors, 4,000 non-descriptors); 1.3MB (RDF/Turtle dump)
• Licence, terms of use: CC BY-NC-ND

**Classification Social Sciences**

• Version: July 2013 version
• Link to documentation:
  o http://www.gesis.org/unser-angebot/recherchieren/tools-zur-recherche/klassifikation-sozialwissenschaften/
  o API documentation for REST service: http://lod.gesis.org/thesozcl/takearest/apidoc
• Intended use: classification of documents
• Data format: SKOS-XL with proprietary extensions, Linked Data HTML representation, downloadable as PDF
• Approximate size: 14 main classes, 145 subclasses
• Licence, terms of use: N/A

**VariableRefCorpus**
This is the reference corpus for variable detection task with manual annotation of variable mentions. It is a compilation of positive and negative samples for referencing variables in Social Sciences publications.

• Link to specification and download: Details of the annotation and guidelines will be delivered with the final release.
• Intended use: this data will be used to train machine learning classifiers to identify segments of paper text that mention variables from social sciences questionnaires.
• License: The corpus contains documents with various open and closed licenses. However, only documents having a CC-BY licence can be shared.

**KeywordRefCorpus**
This is the reference corpus for the keyword assignment task. The documents in this collection are tagged with keywords from TheSoz knowledge base.

• Link to specification and download: Details of the annotation and guidelines will be delivered with the final release.
• Intended use: this data will be used to train machine learning classifiers to assign relevant keywords to papers.
• License: The corpus contains documents with various open and closed licenses. However, only documents having a CC-BY licence can be shared.

**NERRefCorpus**
This is the reference corpus for named entity recognition task. The documents in this corpus have been annotated with named entity tags tailored for social sciences domain.

• Link to specification and download: Details of the annotation and guidelines will be published at https://github.com/openminted/uc-tdm-socialsciences.
• Intended use: Machine learning classifiers will be trained using this data which will be used to annotate named entities in new documents.
• License:
  o Creative Commons - Attribution
  o Creative Commons - Attribution - ShareAlike
  o Creative Commons - Attribution - NonCommercial
  o Creative Commons - Attribution - NonCommercial – ShareAlike
  o Free Digital Peer Publishing License
External Resources

WordNet
- Version: WordNet version 3.0
- Link: https://wordnet.princeton.edu/wordnet/download/current-version/
- License: WordNet license
- Intended use: for feature extraction in variable mention detection task

GermaNet
- Version: GermaNet release 12.0
- Link: http://www.sfs.uni-tuebingen.de/GermaNet/
- License: http://www.sfs.uni-tuebingen.de/GermaNet/licenses.shtml. The resource is used internally under an Academic Research License Agreement granted to GESIS
- Intended use: for feature extraction in variable mention detection task

DBnary
- Link: http://kaiko.getalp.org/about-dbinary/
- Download: http://kaiko.getalp.org/about-dbinary/download/
- License: Dbinary is distributed under Creative Commons Attribution-ShareAlike 3.0
- Intended use: for feature extraction in variable mention detection task

10.1.3 Tools/Components

CorpusCreator
- Version: N/A
- License: N/A (IPR: GESIS)
- Link to documentation: https://github.com/infolis/corpus-creation/blob/master/README.md
- Link to the code: https://github.com/infolis/corpus-creation
- Performed task: given the harvested metadata and content from SSOAR, this component converts paper metadata and creates a subcorpus of documents according to some custom defined criteria
- Input/Output formats:
  - Input: Directory of harvested files and metadata
  - Output: A subset of harvested documents
- Deployment mode: standalone
- Requirements for system environment (OS, installed libs, language interpreters): Python 2.7
- Performance requirements and limits (used memory, processing speed, multi-threading): N/A
- Used static resources and models: none
PdxXmlCreator

- Version: 1.0.1-SNAPSHOT
- License: Apache License v2.0
- Link to the code: https://github.com/openminted/uc-tdm-socialsciences/tree/master/ss-io-pdf
- Performed task: communicates with PDFX service to convert PDF files to XML format
- Input/Output formats:
  - Input: PDF
  - Output: XML (with the pdfx schema)
- Deployment mode: workflow component
- Requirements for system environment (OS, installed libs, language interpreters): JRE 1.8
- Performance requirements and limits (used memory, processing speed, multi-threading): maximum file size 5 MBs and maximum article length 100 pages
- Used static resources and models: none

PdxXmlReader

- Version: 1.0.1-SNAPSHOT
- License: Apache License v2.0
- Link to the code: https://github.com/openminted/uc-tdm-socialsciences/tree/master/ss-io-pdf
- Performed task: Converts the XML output of Pdxf into UIMA XMI format
- Input/Output formats:
  - Input: XML (with the pdfx schema)
  - Output: UIMA JCAS
- Deployment mode: workflow component
- Requirements for system environment (OS, installed libs, language interpreters): JRE 1.8
- Performance requirements and limits (used memory, processing speed, multi-threading): N/A
- Used static resources and models: none

BinaryCasToTsvConverter

- Version: 1.0.1-SNAPSHOT
- License: GNU GPL v3.0
- Link to the code: https://github.com/openminted/uc-tdm-socialsciences/tree/master/ss-module-ner
- Performed task: Converts training data from binary CAS format (as e.g. exported from WebAnno) to TSV format needed by the StanfordNERTrainer (see below)
• Input/Output formats:
  o Input: CAS
  o Output: TSV

• Deployment mode: workflow component
• Requirements for system environment (OS, installed libs, language interpreters): JRE 1.8
• Performance requirements and limits (used memory, processing speed, multi-threading): N/A
• Used static resources and models: none

StanfordNERTrainer
• Version: 1.0.1-SNAPSHOT
• License: GNU GPL v3.0
• Link to documentation: https://builds.openminted.eu/job/uc-socialsciences/ws/ss-doc/target/generated-docs/user-guide.html
• Link to the code: https://github.com/openminted/uc-tdm-socialsciences/tree/master/ss-module-ner
• Performed task: trains a new NER model with training data
• Input/Output formats:
  o Input: file with training data in .tsv format
  o Output: output file for the serialized model
• Deployment mode: workflow component
• Requirements for system environment (OS, installed libs, language interpreters): JRE 1.8
• Performance requirements and limits (used memory, processing speed, multi-threading): N/A
• Used static resources and models: NerRefCorpus

NER-Pipeline
• Version: 1.0.1-SNAPSHOT
• License: GNU GPL v3.0
• Link to documentation: https://builds.openminted.eu/job/uc-socialsciences/ws/ss-doc/target/generated-docs/user-guide.html
• Link to the code: https://github.com/openminted/uc-tdm-socialsciences/tree/master/ss-module-ner
• Performed task: applies a NER model to un-annotated data
• Input/Output formats:
  o Input: input data to be labeled (has to be in XMI (UIMA) format)
  o Output: labels for all recognized Named Entities
• Deployment mode: workflow component
• Requirements for system environment (OS, installed libs, language interpreters): JRE 1.8
• Performance requirements and limits (used memory, processing speed, multi-threading): N/A
• Used static resources and models: NER model
PerformanceMeasure
- Version: 1.0.1-SNAPSHOT
- License: GNU GPL v3.0
- Link to the code: https://github.com/openminted/uc-tdm-socialsciences/tree/master/ss-module-ner
- Performed task: evaluates the results of NER and reports precision and recall scores
- Input/Output formats:
  - Input: gold data with correct labels, prediction data labeled by a NER algorithm (both in XMI format)
  - Output: agreement scores as well as precision and recall
- Deployment mode: workflow component
- Requirements for system environment (OS, installed libs, language interpreters): JRE 1.8
- Performance requirements and limits (used memory, processing speed, multi-threading): N/A
- Used static resources and models: none

DKPro Core Hyphenation Remover
- Version: DKPro Core Version 1.8.0
- License: Apache Software License (ASL) version 2
- Link to documentation: https://dkpro.github.io/dkpro-core/releases/1.8.0/docs/component-reference.html#engine-HyphenationRemover
- Performed task: Fixes hyphenations resulting from pdf to text conversion
- Input/Output formats:
  - Input: UIMA JCAS
  - Output: UIMA JCAS
- Deployment mode: workflow component
- Requirements for system environment (OS, installed libs, language interpreters): JRE 1.8
- Performance requirements and limits (used memory, processing speed, multi-threading): N/A
- Used static resources and models: word dictionary files (English and German dictionary provided)

DKPro Similarity
- Version: 2.2.0
- License: Apache Software License (ASL) version 2 and GNU GPL v3.0
- Link to documentation: https://dkpro.github.io/dkpro-similarity/
- Link to the code: https://github.com/dkpro/dkpro-similarity
• Performed task: Feature extraction in variable mention detection task
• Deployment mode: software library

Weka
• Version: 3.8.0
• Link to resource: http://www.cs.waikato.ac.nz/ml/weka/downloading.html
• Intended Use: Classification of instances in variable detection task
• License, terms of Use: issued under the GNU General Public License
• Deployment mode: software library

JWI (the MIT Java Wordnet Interface)
• Version: 2.4.0
• Link: https://projects.csail.mit.edu/jwi/
• License: The software is distributed under a license that makes it free to use for all purposes, as long as proper copyright acknowledgement is made.
• Performed task: Feature extraction in variable mention detection task
• Deployment mode: software library

JMWE (Java library for detecting Multi-Word Expressions in text)
• Version: 1.0.2
• Link: JMWE : http://projects.csail.mit.edu/jmwe/
• License: CC-BY 4.0
• Performed task: Feature extraction in variable mention detection task
• Deployment mode: software library

JavaAPI for GermaNet
• Version: 9.0.3
• Link: http://www.sfs.uni-tuebingen.de/GermaNet/tools.shtml
• Performed task: Feature extraction in variable mention detection task
• Deployment mode: software library

Stanford NamedEntityRecognizer
• Version: 3.8.0
• Link: https://stanfordnlp.github.io/CoreNLP/
• Software: https://nlp.stanford.edu/software/CRF-NER.html
• License: GNU GPL v3.0
• Performed task: used inside NER module for training a NER model
• Deployment mode: software library

**METEOR**

- Version: meteor version 1.5
- Link: http://www.cs.cmu.edu/~alavie/METEOR/
- Software: http://www.cs.cmu.edu/~alavie/METEOR/index.html#Download
- License: released under the GNU Lesser General Public License (LGPL)
- Performed task: used for evaluation of variable mention detection component
- Deployment mode: software library

**TreeTagger**

- Version: 3.2
- License: TreeTagger 3.2: “freely available for research, education and evaluation. Usage of the system for commercial purposes is forbidden.”
- Link to documentation: http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/ (original doc)
- Link to the code: Executable code (not open source) available at http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/
- Performed task: Part-of-Speech tagging and lemmatization in Segmenter-POS-Tagger
- Used static resources and models: Parameter files available for several languages; German and English parameter file is used in this use case.

10.1.4 Services

**pdfx**

- Address: http://pdfx.cs.man.ac.uk/
- Performed Tasks:
  - Pdf to xml conversion (taking care of the structure of scientific articles, i.e. marking sections, abstract, references etc.)
  - Sentence splitting (using another service called Punkt)
  - Extracts figures and stores them as image files
  - Resolves DOIs for references
- Input/Output formats:
  - Input: pdf
  - Output: xml (+png for images/figures)
- Terms of use (esp. performance limitations):
  - Max. 5MB file size for input pdf
  - Max. 100 pages for input pdf
  - Less than 1000 requests per job (for larger requests should contact the system administrators)
- Hosting institution: Advanced Interfaces Group (AIG) at University of Manchester
• Type of interface (REST/SOAP/Other): via web browser or other HTTP client, e.g. curl

10.1.5 Other

ZACAT

• Version: Nesstar Server 4.0
• Link to resource (url): http://zacat.gesis.org/
• Link to documentation:
  o Dataset server is an instance of NESSTAR server whose documentation can be found at http://www.nesstar.com/help/4.0/server/index.html
• Intended use: to be used in the subtask of variable detection (look up the variables in a study)
• Data format: various formats, including RDF, Java objects or REST service responses
  o RDF: http://zacat.gesis.org/obj/fCatalog/ZACAT@datasets
• Approximate size (MBs, number of entities/classes): about 670 datasets, with more than 250,000 variables in total
• Licence, terms of use: free access

10.2 Deployment Plan

The TDM solution is a preprocessing step that feeds data into the information retrieval (IR) system hosted by GESIS. This TDM system is required to provide entities and links between them for the IR system. In our use-case we are interested in three different classes of entities; (1) named entity mentions in full-texts, (2) dataset and dataset variable mentions in full-texts (occurring only in empirical studies), and (3) topical keywords that are relevant for each article.

Named entity mentions in the full-texts will be detected and added to paper metadata so the IR system can use them to improve search results. Dataset and variable mentions inside the text will also be added to the metadata and link to corresponding data on dataset archive (ZACAT). Finally, the extracted keywords will be added to “Keywords” metadata fields to improve information retrieval.

GESIS expects to run the TDM processes on premise for production purposes. They will instantiate the workflow on their infrastructure and run it internally on a scheduled basis (either timed schedule or triggered schedule which will be activated by for instance uploading a new paper to SSOAR).

10.3 Data Interfaces

SSOAR crawler

• Type of interface: OAI-PMH interface
• Protocol: OAI-PMH
• Format of data transferred: PDF and XML (for metadata)
• Level of automation: high
PDFX web interface
- Type of interface: web interface
- Protocol: http
- Format of data transferred: PDF, XML, HTML, .tar.gz
- Level of automation: high

ZACAT interface
- Type of interface: web interface
- Protocol: HTTP
- Format of data transferred: RDF XML
- Level of automation: medium

TheSoz interface
- Type of interface: web interface
- Protocol: REST api
- Format of data transferred: RDF/XML, RDF/Turtle, RDF/NTriples
- Level of automation: medium

TheSoz classification interface
- Type of interface: web interface
- Protocol: REST api
- Format of data transferred: RDF/XML
- Level of automation: medium

10.4 User Interfaces

WebAnno
Using WebAnno, social sciences experts can annotate named entities in full-texts. The annotated data will be used as training and evaluation data for training named entity recognition models and testing them.

GESIS-wide search
GWS will be (currently under development by GESIS, not in context of OpenMinTeD) the central search entry point at GESIS where people can search for scientific publications, projects, datasets and people in the broad field of Social Sciences. It is also the place where the linked information from the LOD
backend is used, e.g. to display related datasets in the detail view of a publication. Our results will be shown here to the user, e.g. references variables in a publication.

### 10.5 Data Processing Scenarios

The data processing consists of four main steps; namely data harmonization, named entity recognition, keyword assignment and dataset and variable mention detection. These steps and the process workflow are depicted in figure below.

![Data Processing Workflow Diagram]

In data harmonization step, metadata of articles (XML) and article contents (PDF) are retrieved from SSOAR repository. Content is then converted to machine readable format (UIMA XMI in our case) which are then stored in internal datastore at GESIS. The XMI version of articles along with the metadata are used in further processing steps. Having the converted version of documents, text analysis components (named entity recognition, keyword assignment and dataset and variable mention detection) can be run on the articles. Each of these annotation models load their classifier models from pre-trained files and annotate phrases inside the text (named entity recognition and
variable mention detection) or add annotations to the document. The document with new annotations and updated metadata is written to the datastore and GESIS IR system indexes will be updated accordingly.

**NER Processing Scenario:**

For the Named Entity Recognition task an entire processing pipeline has been implemented. It can be performed on own data, in the case of the Social Science use case to scholarly Social Science publication to extract relevant entities such as authors, locations, institutions, journals, publishers. In general the pipeline allows to input some annotated data to train an own custom model, to apply a trained model to new un-annotated data, and evaluate the performance of any model. The pipeline consists of the following steps (see https://builds.openminted.eu/job/uc-socialsciences/ws/ss-doc/target/generated-docs/user-guide.html):

**Pre-processing:**

The pre-processing step converts a collection of PDF files to XMI format which can be annotated by WebAnno to be used as training data in the following step.

**Training:**

The annotated training data are then used to train the own custom model, in our case a specific model that fits better to Social Science entities. To do this, the training data has to be in TSV format with one token per line, sentences separated by newlines, and annotations in the second column. Files exported from WebAnno (binary CAS format) should be converted to TSV format using a particular converter developed for this purpose. As model trainer the StanfordCoreNLP trainer has been used.

**Recognition:**

The own NER model is then applied to un-labeled, such that the output will contain labels for all of the recognized use case specific Named Entities. Again, input data has to be in XMI (UIMA) format, so PDF files have to be converted beforehand. Instead of the own NER model, pre-trained models that are openly available can be used as well (but mind that those models are mostly trained on newswire text, so if applied to a different domain, the results may not have the quality expected).

**Evaluation:**

To evaluate the results of the NER process a particular program has been developed which needs gold data, i.e. manually annotated data with the correct NE labels, as input as well as the prediction data, i.e. documents annotated with the NER. Both have to be in XMI format again. The program will output agreement scores as well as precision and recall.

Our components are currently command-line applications, because they are intended to be incorporated into GESIS services. So in the end, they will run on a server in background and feed the results in databases, indexes or the like.
Variable Detection Processing Scenario:
For the Variable Detection task a first workflow prototype has been developed and evaluated. The processing workflow consists of the following steps:

**Document Retrieval:**
A pre-defined set of documents which contain one or more references to survey studies (results of InFoLiS algorithm or bibliographies) is retrieved.

**Pre-processing:**
All documents are then converted from PDF to XMI format.

**Retrieval of a Set of Survey Variables:**
A set of variables that belong to the survey studies are ingested from a ZACAT xml data dump. This variable set is then filtered by the topic of the documents (which is included in the documents’ metadata)

**Query Transformation:**
The variable question type is determined (e.g. Variables staring with “How often..” are Frequency-Questions)

**Sentence Retrieval:**
The documents are split into sentences and as an optional step specific passages of the document may be neglected (e.g. survey variables generally do not occur in the abstract, references)

**Linguistic Processing:**
Features from variables and sentences in the document are extracted using State-of-the-art tools for tokenizing, lemmatizing, PoS tagging, NER, keyword extraction (using TheSoz), WordNet & GermaNet Expansion, etc.) and the pair-wise similarity scores is computed (using Dkpro Similarity, Meteor, etc.)

**Classification - Machine Learning:**
(Train) and assign the variable class by running (a set of) supervised ML classifiers.

**Annotation:**
Potential references to variables (sentences from document) and their variable IDs are returned along with a confidence score. Optional: References to variables are annotated by high lightening in the document.

**Evaluation:**
Evaluate output of algorithm on split train/dev/test set.

### 10.6 Limitations
The first limitation is that we are relying on a web service (pdfx) for pdf to xml conversion. We chose it because it was performing best among all pdf to text converters that we tested. But pdfx web service is unreliable as it may stop working any time (which we also already experienced during testing).
Moreover, since our system has tight coupling with internals of this web service, it is prone to failure if some internal communication protocols of the service change. On the other hand, we have the original pdfx source code at our hands as the developers provided it, but there is currently no licence agreement attached to the code. This makes it impossible to redistribute the code along with our system; hence, causing difficulty for adoption and reuse of our system by others. The other problem with this web service is that it has a limitation on file size (can be turned off when running locally) and article length, and also on the number of requests (depends on their server infrastructure). To overcome this limitation, we have implemented a new converter based on CERMINE PDF reader.

The other limitation we are currently facing is that even open access articles may have licences that do not permit redistribution or distribution of modified content (annotated articles in our case). From the social sciences open access repository (SSOAR), only 8.62% of a total of c.a. 37000 documents have a licence that allow us to publish our datasets. And if we narrow than the documents to English and German documents (which are concerned in our experiments) this will go down to 6.77% of all of the articles. For our variable mention detection scenario, we need to narrow this subset even further to only empirical publications, which may lead to an insufficient amount of textual data available for this scenario.

10.7 Release Plan

There are three planned releases for the applications. These are as follows:

- Jan 2017 - 1st Application Release
- August 2017 - 2nd Application Release
- Feb 2018 - 3rd Application Release

Consequently, the release plan will be as follows:

1st release (only registry available):

In the first release, we manually annotated a collection sample of social science publications from SSOAR for the named entity recognition task. A new supervised classifier model was trained to adapt it to the domain of Social Science publications. The precision and recall results have been not very good due to the small training set, but we could show the potential of the application. As a result, we have implemented the entire pipeline for the named entity recognition task (see chapter 8.5). In terms of data preparation, which is the essential pre-processing step for all the three subtasks, we have published our pipeline for converting PDF to UIMA XMI. The output XMI files can be used e.g. in WebAnno for annotation, or as input to other UIMA based components. Our first application release is based on UIMA-Fit pipeline. Additionally, having the first annotated dataset available, we are able to evaluate state-of-the-art named entity recognition components and models on our data. Based on this evaluation we can choose the proper component(s) and configure them with our data.
For the second (variable detection) and the third task (keyword assignment), we also started to compile a sample corpus for evaluation.

2nd release (initial version of workflow service available):
For the second task a Gold Standard Corpus for Variable Detection is being built up. The definition of the task and the human annotation procedure will be described in the Annotation Guidelines ‘Linking Variable Mentions in Social Science Publications’. A first prototype of a variable mention detection component will be ready at this stage based on supervised Machine Learning approach trained and tested for a subset of variables and a small corpus of Social Science publications along with evaluation studies (see Zielinski, Andrea, and Peter Mutschke. 2017. "Mining Social Science Publications for Survey Variables." In Proceedings of NLP+CSS: Workshop on Natural Language Processing and Computational Social Science at ACL '17 Vancouver, Canada)

For the keyword extraction task we started a study of off-the-shelf keyword assignment components (MAUI Term Extraction, TextRank, rake) to see how they can be applied to our data. The evaluation has been carried out on approx. 1.000 SSOAR publications and the results will be published soon.

3rd release (annotation viewing + editing service available):
In the third release, a final version of the processing pipeline for all three tasks along with the complete user manual will be delivered. The final system will be implemented based on the OpenMinTeD workflow service. Additionally, a workshop will be held to present the work done in this use-case and encourage researchers in using the developed system. The final system will be ready to be integrated in GESIS infrastructure and also ready to be used and extended by interested researchers.
11. SC-A: Research Analytics – Funding Mining Services

The Funding Mining application mines the fulltext of publications and extracts links to projects. Currently, projects from EC (FP7/H2020), NSF (National Science Foundation, USA), NIH (National Institute of Health, USA), Wellcome Trust, FCT (Fundação para a Ciência e a Tecnologia, Portugal), ARC (Australian Research Council), NHMRC (National Health and Medical Research Council, Australia), CSF/HRZZ (Hrvatska Zaklada Za Znanost, Croatia), MSES-MZOS (Ministarstvo Znanosti, Obrazovanja i športa, Croatia), SFI (Science foundation Ireland), NWO (Nederlandse Organisatie voor Wetenschappelijk Onderzoek, Netherlands) are supported, but new funders are added regularly.

Initially developed as part of OpenAIRE’s Inference (by mining) workflow to enrich the OpenAIRE information space, it is actively used as a standalone workflow in several E.U. funding & research evaluation tenders providing useful insight and timely intelligence of systematic research in E.U.

11.1 Resources

11.1.1 Document Formats

<table>
<thead>
<tr>
<th>Name</th>
<th>Specification</th>
<th>Intended Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utf-8 encoded JSON</td>
<td><a href="http://www.json.org">http://www.json.org</a></td>
<td>Text Mining</td>
</tr>
<tr>
<td>Utf-8 encoded XML</td>
<td><a href="http://www.w3.org/XML/">http://www.w3.org/XML/</a></td>
<td>Text Mining</td>
</tr>
<tr>
<td>Utf-8 encoded Plain text</td>
<td></td>
<td>Text Mining</td>
</tr>
</tbody>
</table>

11.1.2 Knowledge Bases

Funders project lists are used as the knowledge base. They are retrieved either via funders’ APIs or they are provided offline in Excel files. They are stored in an SQLite database and they are almost 1GB. These lists contain the values of the following attributes:

- Funder Name (i.e. EC, NSF, NHMRC, etc.)
- Project Identifier (mandatory)
- Project Acronym (if it exists - some funders do not have this)
- Project Title (optional)
- Project start/end date (optional)

The license may vary according to the funder, but all these lists are open to use for text mining.

11.1.3 Tools/Components

Project mining module
Name: madIS funding mining  
Version: 1.0  
License: Apache License 2.0  
Documentation: https://github.com/johnfouf/FundingMining_OMTD  
Source code: https://github.com/johnfouf/FundingMining_OMTD  
Performed task: Project linking  
Input format: XML  
Output format: JSON  
Deployment mode: Executable  

11.1.4 Services
The application does not use any external web services.

11.2 Deployment Plan
This application is already deployed on OpenAIRE infrastructure (https://www.openaire.eu/).

The tool will be deployed as a docker executable in a virtual machine VM. The required infrastructure is provided by the Greek Research and Technology Network (https://grnet.gr/en/) through the cloud service OKEANOS (https://okeanos.grnet.gr/home/). All the resources (i.e. Knowledge bases, documents, etc.) are stored in the same VM.

11.3 Data Interfaces

<table>
<thead>
<tr>
<th>Type of interface</th>
<th>Documents Interface</th>
<th>Projects Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>Command Line Tool</td>
<td>Database (SQLite)</td>
</tr>
<tr>
<td>Format of data transferred</td>
<td>XML (given in the standard input)</td>
<td>SQL</td>
</tr>
<tr>
<td>Level of automation</td>
<td>Plain Text</td>
<td>Structured (SQLite table)</td>
</tr>
<tr>
<td></td>
<td>Automated</td>
<td>Automated</td>
</tr>
</tbody>
</table>

11.4 User Interfaces

Although the application is implemented as a docker executable for better performance, it also interacts with users through the http://mining.openaire.eu/openaireplus/analyze interface. Users may upload their utf-8 encoded text and get the results in JSON.
### 11.5 Data Processing Scenarios

<table>
<thead>
<tr>
<th>Stage</th>
<th>Title</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data input</td>
<td>The module gets document’s plain text in XML format, and project information that are stored in an SQLite DB.</td>
</tr>
<tr>
<td>2</td>
<td>Processing</td>
<td>Text mining phase: Extracts project links using the madIS funding mining module.</td>
</tr>
<tr>
<td>3</td>
<td>Data output</td>
<td>The extracted project links are stored in JSON files.</td>
</tr>
</tbody>
</table>

#### 11.5.1 Application Updates
The application is updated when a new funder is included. This happens because we tune the algorithm to mine the new funder’s projects with higher accuracy. This is a manual process which will happen every 2 months.

#### 11.5.2 Resource Updates
Project lists need to be periodically updated. The update requires some manual work to download and insert the new projects (or the new funders) in the existing SQLite DB. This will happen every 2 months.

#### 11.6 Limitation
The application works only with UTF-8 encoded text. There are no limitations regarding memory usage, scalability or anything else.

#### 11.7 Release Plan

**11.7.1 January 2017**
Integration of madIS funding mining in OMTD workflow Engine
12. **SC-B: Research Publications Recommendation System**

This application offers content-based recommendations to scholarly materials, such as related research papers. The recommender relies on metadata, full text features and other features that can be mined from the article full texts or are obtained from external sources, including citations, readership, affiliations, venues, publication year, etc.

### Resources

#### 12.1.1 Document Formats

<table>
<thead>
<tr>
<th>Name</th>
<th>Link</th>
<th>Intended use</th>
<th>License, version, serialisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Javascript Object Notation (JSON)</td>
<td><a href="http://www.json.org/">http://www.json.org/</a></td>
<td>Serialisation format for data object exchange</td>
<td>N/A</td>
</tr>
<tr>
<td>Extensible Markup Language (XML)</td>
<td><a href="https://www.w3.org/XML/">https://www.w3.org/XML/</a></td>
<td>Markup language for metadata description</td>
<td>N/A</td>
</tr>
<tr>
<td>Portable Document Format (PDF)</td>
<td><a href="http://www.adobe.com/content/dam/Adobe/en/dev">http://www.adobe.com/content/dam/Adobe/en/dev</a></td>
<td>Extract text features</td>
<td>N/A</td>
</tr>
</tbody>
</table>

#### 12.1.2 Knowledge Bases

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>Link to resource</th>
<th>Link to documentation</th>
<th>Intended use</th>
<th>Data format</th>
<th>Approximate size</th>
<th>License, terms of use</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORE dataset</td>
<td>2.0</td>
<td><a href="http://core.ac.uk/docs#dataset">http://core.ac.uk/docs#dataset</a></td>
<td><a href="http://core.ac.uk/docs#dataset">http://core.ac.uk/docs#dataset</a></td>
<td>Publications data</td>
<td>Json, pdf</td>
<td>As of August 2017</td>
<td>CC-BY 4.0</td>
</tr>
<tr>
<td>METADATA</td>
<td>SCHEMA</td>
<td>DESCRIPTION</td>
<td>LICENSE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>-------------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMTD-SHARE metadata schema</td>
<td>2.0</td>
<td><a href="https://guidelines.openmined.eu/the_omtd-share_metadata_schema.html">https://guidelines.openmined.eu/the_omtd-share_metadata_schema.html</a></td>
<td>Richer description of metadata</td>
<td>CC-BY 4.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dublin CORE</td>
<td></td>
<td><a href="http://dublincore.org/">http://dublincore.org/</a></td>
<td>Vocabulary of bibliographic terms</td>
<td>DC MI Public License 2.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RIOX Metadata Application Profile</td>
<td>2.0</td>
<td><a href="http://rioxx.net/">http://rioxx.net/</a></td>
<td>Metadata format of UK based institutional repositories</td>
<td>CC-AS 3.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
12.1.3 Tools/Components

<table>
<thead>
<tr>
<th>Name</th>
<th>PdfBox</th>
<th>Parscit</th>
<th>Grobid</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>2.0.7</td>
<td>130908</td>
<td>0.4.1</td>
</tr>
<tr>
<td><strong>License</strong></td>
<td>Apache License 2.0</td>
<td>GNU Lesser General Public</td>
<td></td>
</tr>
<tr>
<td><strong>Task performed</strong></td>
<td>Extract plain text from binary pdf</td>
<td>Extract citations from scientific text</td>
<td>Extract semantic sections from binary pdf</td>
</tr>
</tbody>
</table>
| **I/O format**   | Input: binary pdf  
Output: plain text | Input: plain text  
Output: annotated xml with references | Input: binary pdf  
Output: annotated xml (tei format) |
| **Deployment mode** | Java library | Standalone application (invoked with RPC) | Java library, Static resources need to be hosted in the same machine |
| **System requirements** | Java Virtual Machine | CRF++ (https://taku910.github.io/crffp) | Java Virtual Machine |
| **Performance requirements** | Processing of 1 publication per second per worker | Processing of 2 publications per second per worker | Processing of 1 publication per second per worker |
| **Static resources** | - | Trained models of bigrams, section labels, func words, keywords,... | cybozu.labs Cybozu labs detection  
Wipo analyser  
wapiti sequence labelling toolkit  
CRF++ |

<table>
<thead>
<tr>
<th>Name</th>
<th>ImageMagik</th>
<th>Lucene/Elasticsearch</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>7.0</td>
<td>6.6</td>
</tr>
<tr>
<td><strong>License</strong></td>
<td>Apache 2.0</td>
<td>Apache 2.0</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Task performed</td>
<td>Generate previe images from pdfs</td>
<td>Analyse, Vectorise and index textual content</td>
</tr>
<tr>
<td>I/O format</td>
<td>Input: pdf</td>
<td>Input: xml and textual content of pdf</td>
</tr>
<tr>
<td></td>
<td>Output: jpeg image</td>
<td>Output: inverted index</td>
</tr>
<tr>
<td>Deployment mode</td>
<td>Standalone application (invoked via RPC)</td>
<td>Part of ElasticSearch search engine</td>
</tr>
<tr>
<td>System requirements</td>
<td>-</td>
<td>Java Virtual Machine</td>
</tr>
<tr>
<td>Performance requirements</td>
<td>Processing: 1 publication per second per worker</td>
<td>Processing: 100 publications indexed per second</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Search: complex filtered queries response in less than 2 seconds</td>
</tr>
<tr>
<td>Static resources</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### 12.1.4 Services

**DOI resolution (CrossRef)**

<table>
<thead>
<tr>
<th>Service</th>
<th>CrossRef API</th>
<th>Mendeley Catalogue API</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td><a href="http://api.crossref.org">http://api.crossref.org</a></td>
<td><a href="http://dev.mendeley.com/methods/">http://dev.mendeley.com/methods/</a></td>
</tr>
<tr>
<td>Task performed</td>
<td>DOI resolution</td>
<td>Readership of publication</td>
</tr>
<tr>
<td>I/O format</td>
<td>Input: open url query of title/authors of publication</td>
<td>Input: DOI of publication</td>
</tr>
<tr>
<td></td>
<td>Output: json response</td>
<td>Output: json response</td>
</tr>
<tr>
<td>Terms of use</td>
<td>“CrossRef asserts no claims of ownership to individual items of”</td>
<td><a href="https://www.mendeley.com/terms">https://www.mendeley.com/terms</a></td>
</tr>
</tbody>
</table>
bibliographic metadata and associated Digital Object Identifiers (DOIs) acquired through the use of the CrossRef Free Services

<table>
<thead>
<tr>
<th>Hosting institution</th>
<th>Crossref</th>
<th>Mendeley (Elsevier)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface type</td>
<td>REST</td>
<td>REST</td>
</tr>
</tbody>
</table>

12.2 Deployment Plan

Parts of this application is already deployed in CORE infrastructure ([https://core.ac.uk/services#recommender](https://core.ac.uk/services#recommender)). In this production release, priority was given to stability, therefore some of the features described in section 1.5 are omitted. We plan to utilise OpenMinted infrastructure to further develop this application and incorporate new developed features to the production service. Namely OMTD workflows are to be used for implementing data processing scenarios 3 and 4.

12.3 Data Interfaces

Communication with the recommender is done via a simple REST Web service. The input metadata are passed as form parameters and response is a semi-structured Json.

12.4 User Interfaces

Research publication recommender system is made available via an embeddable Javascript widget. Widget is responsible to gather metadata in the webpage that is present and invoke the web service described above. It then parses the response and renders a visually appealing html output. The widget is included in display pages of CORE.
Also the recommender widget is embedded in external repositories display page with the extra feature of filtering to provide recommendations coming from that repository.

Year: 2005  
**OAI identifier:** oai:oro.open.ac.uk:112  
**Provided by:** Open Research Online

**Downloaded from** [http://oro.open.ac.uk/112/1/ISDRC_Helsinki_05_v11.pdf](http://oro.open.ac.uk/112/1/ISDRC_Helsinki_05_v11.pdf)

### Suggested articles

**Economic analysis of World Bank education projects and project outcomes**  
**Provided by:** Research Papers in Economics  
**By:** Yamda Ayesha Yaqub, Mooock Peter, Gittinger J. Price, Patrinos Harry...

**The Use of System Dynamics Simulation Models in Project Management Education**  
**Provided by:** Sunderland University Institutional Repository  
**By:** Ahmed Heba Saleh

**Muslim Pupils, Children's Fiction and Personal Understanding**  
**Provided by:** University of Worcester Research and Publications  
**Publisher:** Shah Abdul Latif University, Khairpur Sindh, Pakistan.  
**By:** Gilani-Williams F., Bigger Stephen

**The uptake and implementation of sustainable construction: Transforming policy into practice**

### 12.5 Data Processing Scenarios

**Communication**
The recommender takes as an input a reference document which is represented as a set of features using the Vector Space Model. These features include the identifier, title, authors, abstract and publication year.

Text mining:

1. Identification: Match input document to CORE corpus either via exact match of identifier or fuzzy matching (above a confidence threshold) of other features.
2. If step 1 is successful (i.e. input document exists already in CORE corpus) make use of richer features like raw text, publisher, journal, author’s affiliations, etc.
3. Infer extra features, e.g.:
   3.1.1 language detection
   3.1.2 significant term extraction
   3.1.3 entities extraction
4. If identifier used is DOI then external services are used to fetch further metadata, e.g.:
   4.1 Readership from Mendeley Catalogue API
   4.2 Metadata enrichment via CrossRef API
5. Construct weighted boolean query to lucene index and filter to provide high quality results. Along with different weight to features described above, we also boost (rank higher certain publications). Boostings include:
   5.1 Provide boost to recent publications by applying standard exponential decay function
   5.2 Negative boosting to metadata-only publications
   5.3 Negative boosting to no-preview publications

Workflow

Reference document is provided and recommendation results is provided to user. If user clicks on certain result this is recorded as an event. Internal model (step 5 – constructing the weighted Boolean query) is retrained periodically after evaluation of online data (form of online machine learning)

Storage publish

*Results of a given input are stored in an internal cache to be served faster in future queries. Cached results are invalidated periodically (to be kept updated with corpus updates).*

12.6 Limitations

The application relies on the availability of full text of publications. In order for the widget to operate correctly it requires the presence of specific meta tags (that are picked from the widget to include them in the web service invocation)
12.7 Release Plan

1.0 January 2017 Beta version of recommender released based exclusively on CORE data

2.0 January 2018 Fully operational recommender based on OpenMinTED corpus (built via Openminted registry)
13. **SC-C: Research Performance**

Application to track research performance (citations, downloads, altmetrics, semantometrics) of papers, individuals, organisations, etc. delivered in the form of a dashboard. The application relies on access to the full texts and metadata to be able to slice and dice the data and extract full text features indicative of performance (for semantometrics). The application then displays graphs allowing to perform benchmarking, comparison and trend analysis with regards to content and performance.

**Resources**

13.1.1 Document Formats

CORE dataset (articles’ metadata described in json format, full text pdfs) and *Openminted corpus (described in OMTD-SHARE schema). Documents from both corpuses are available in the following formats:*

<table>
<thead>
<tr>
<th>Name</th>
<th>Link</th>
<th>Intended use</th>
<th>License, version, serialisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Javascript Object Notation (JSON)</td>
<td><a href="http://www.json.org/">http://www.json.org/</a></td>
<td>Serialisation format for data object exchange</td>
<td>N/A</td>
</tr>
<tr>
<td>Extensible Markup Language (XML)</td>
<td><a href="https://www.w3.org/XML/">https://www.w3.org/XML/</a></td>
<td>Markup language for metadata description</td>
<td>N/A</td>
</tr>
<tr>
<td>Portable Document Format (PDF)</td>
<td><a href="http://www.adobe.com/content/dam/Adobe/en/dev">http://www.adobe.com/content/dam/Adobe/en/dev</a></td>
<td>Extract text features</td>
<td>N/A</td>
</tr>
</tbody>
</table>

13.1.2 Knowledge Bases

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>Link to resource</th>
<th>Link documentation to documentation</th>
<th>Intended use</th>
<th>Data format</th>
<th>Approximate size</th>
<th>License, terms of use</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORE data</td>
<td>2.0</td>
<td><a href="http://core.ac.uk/docs#dataset">http://core.ac.uk/docs#dataset</a></td>
<td><a href="http://core.ac.uk/docs#dataset">http://core.ac.uk/docs#dataset</a></td>
<td>Publication</td>
<td>Json</td>
<td>As of August</td>
<td>CC-BY 4.0</td>
</tr>
<tr>
<td>Schema</td>
<td>Version</td>
<td>Source 1</td>
<td>Source 2</td>
<td>Data Source</td>
<td>Metadata</td>
<td>Format</td>
<td>License</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------</td>
<td>----------------------------------------------</td>
<td>----------------------------------------------</td>
<td>-------------------------------------------------</td>
<td>----------</td>
<td>-----------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Public CORE</td>
<td></td>
<td><a href="http://dublincore.org/">http://dublincore.org/</a></td>
<td><a href="http://dublincore.org/">http://dublincore.org/</a></td>
<td>Vocabulary of bibliographic terms</td>
<td>xml</td>
<td>-</td>
<td>DCMI Public License 2.0</td>
</tr>
<tr>
<td>RIOXX Metadata Application Profile</td>
<td>2.0</td>
<td><a href="http://rioxx.net/">http://rioxx.net/</a></td>
<td><a href="http://rioxx.net/">http://rioxx.net/</a></td>
<td>Meta data format of UK based institutions</td>
<td>xml</td>
<td>-</td>
<td>CC-AS 3.0</td>
</tr>
</tbody>
</table>

---

**Public** Page 145 of 229
<table>
<thead>
<tr>
<th>Repository</th>
<th>API Name</th>
<th>API URL</th>
<th>API URL</th>
<th>Data Format</th>
<th>Data Volume</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossref API</td>
<td></td>
<td><a href="http://api.crossref.org/">http://api.crossref.org/</a></td>
<td><a href="http://api.crossref.org/">http://api.crossref.org/</a></td>
<td>JSON</td>
<td>86 million publications</td>
<td>Crossref specific - &quot;CrossRef asserts no claims of ownership to individual items of bibliographic metadata and associated Digital Object Identifiers (DOIs) acquired through the use of the CrossRef Free Services</td>
</tr>
<tr>
<td>Altmetrics API</td>
<td></td>
<td><a href="https://api.altmetric.com/">https://api.altmetric.com/</a></td>
<td><a href="https://api.altmetric.com/">https://api.altmetric.com/</a></td>
<td>JSON</td>
<td></td>
<td>Commercial license if data used in commercial application</td>
</tr>
</tbody>
</table>
### 13.1.3 Tools/Components

<table>
<thead>
<tr>
<th>Name</th>
<th>Python pandas</th>
<th>NLTK</th>
<th>Scikit-learn</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>0.20.3</td>
<td>3.2.4</td>
<td>0.19.0</td>
</tr>
<tr>
<td><strong>License</strong></td>
<td>BSD</td>
<td>Apache 2.0</td>
<td>BSD</td>
</tr>
<tr>
<td><strong>Task performed</strong></td>
<td>Describe publications data in dataframes to further execute mining functions (e.g. aggregations, slicing, merging)</td>
<td>Execute basic NLP functions (tokenization, lemmatisation, POS tagging, segmentations, etc.)</td>
<td>Train basic ML models to perform classification (using SVM classifier, extract text features)</td>
</tr>
<tr>
<td><strong>I/O format</strong></td>
<td>Input: publication metadata</td>
<td>Output: raw text</td>
<td>Input: corpus of publications</td>
</tr>
<tr>
<td></td>
<td>Output: Pandas dataframe</td>
<td>Output: processed and annotated text</td>
<td>Outputs: classified data</td>
</tr>
<tr>
<td><strong>Deployment mode</strong></td>
<td>Essential element of larger python workflow</td>
<td>Part of python workflow</td>
<td>Part of python workflow</td>
</tr>
<tr>
<td><strong>System requirements</strong></td>
<td>Python v3</td>
<td>Python v3</td>
<td>Python v3</td>
</tr>
<tr>
<td><strong>Performance requirements</strong></td>
<td>Memory to hold dataframes up to 10000 publications</td>
<td>Process 5 publications per second per worker</td>
<td>Training: training new model within an hour</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Predict (classify) 10 publications per second per worker</td>
<td></td>
</tr>
<tr>
<td><strong>Static</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
13.2 Deployment Plan

A production instance of the application will be deployed in the CORE infrastructure http://analytics.core.ac.uk.

13.3 Data Interfaces

The application data operations can be split in three main categories:

**Microservices to collect backend data**: These are light services that run as daemons and query external web services to fetch all available data. Data are stored in raw format in files or document store databases.

**Post fetch services**: Micro data services responsible to process, store and index raw data fetched by the microservices

**REST interfaces**: These are internal APIs that expose indexed data to the frontend UI using a light yet structured data format (e.g. Json)

13.4 User Interfaces

The web interface will offer to users the following:

- the ability to explore scientific publications according to various dimensions (e.g. by institution, year, author)
- to rank papers according to their metrics
- dive into authors top publications
- rank publications according to various criteria. These can be:
  - most important citations
  - most interactions in social media
  - best readership
- Ability to compare impact of publications sets (e.g. publications of two different universities or publications of two different authors)

13.5 Data Processing Scenarios

1. DOI discovery: Match input document to CORE corpus either via exact match of identifier or fuzzy matching (above a confidence threshold) of other features
2. For documents that DOI is discovered fetch available metadata from
   a. Mendeley API: we are interested in readership and disciplines of readers
   b. CrossREF API: enrich existing metadata (citations if publisher part of I4OC, affiliations of authors)
   c. Microsoft Academic Search API: fetch citations’ counts, list of references, venues (conferences), journals
   d. Altemetrics: fetch mentions of publication in social media, citation counts, readerships.
3. Store retrieved data in raw format
4. Normalization and harmonization data
5. Term and entities extraction
6. Construct citation graph

13.6 Limitations
The application relies on data fetched via external web services. Therefore, we need to respect the usage quotas of the external web services. Data collected reflect the time they were captured and if not regularly updated can become obsolete and lead into wrong results. Also, only publications with a DOI are considered reducing the available corpus under consideration and therefore the coverage of this tool.

13.7 Release Plan
Version 0.1 (alpha) Jan 2018 First public release of RETE with basic functionalities
Version 1.0 (beta) May 2018 Release of the analytics dashboard prototype
14. **SC-D Text mining of articles related to Leica Microscopes**

This application design document represents Frontiers’ efforts so far and is subject to further updates according to concretization of the application’s requirements.

Today, research results take the form of a publication (i.e. article, data, and code), collectively referred to as article resources. Research results can also produce other forms of output (i.e. laboratory equipment, cell-lines, research protocol) and software.

In this use case, we use Text and Data Mining as a tool to measure the research impact of a Leica microscope.

Programmatically, the TDM steps are as follows:

Step #1: The SME defines the keywords and synonyms when searching for Leica microscopes in the articles. For example, models and synonyms.

Step #2: Leica Microscope TDM program reads the OMTD data repository and extracts all articles where Leica Microscope occurs.

Step #3: Leica Microscope TDM program downloads and stores the Leica Microscope articles.

Step #4: Leica Microscope TDM program uses the annotation tools to annotate the relevant keywords and synonyms.
Step #5: Leica Microscope TDM program stores the annotated articles

Step #6: Leica Microscope TDM program updates the list of articles, with the domain description, institution, year and a link to the annotated article and the original article

14.1 Resources

The Leica Microscope TDM program proof of concept is at the very early stages of requirements and design. Using agile methodology, resources (e.g. document formats, knowledge bases, tools/components, services, and other resources) will be defined as the first proof-of-concept is developed in October.

14.2 Deployment Plan

The first proof-of concept will be a stand-alone program running independently of OMTD using some of its tools. As the OMTD platform and interfaces become available, the TDM programs will be executed from the platform with the specified APIs and components.

This section will be updated as the OMTD platform and interfaces become available.

14.3 Data Interfaces

As of this writing, the main interface is to the CORE and OpenAIRE data repository to extract relevant articles, to develop a table of Research Impact.

The details (e.g. type of interface, protocol, format of data transferred, and level of automation) will be updated as the proof-of-concept matures into a full-blown Leica Microscope TDM program.

14.4 User Interfaces

The details of the User Interface will be designed and developed after the main extract, annotate, define dictionary of features, connectors, and transition terms are available sometime December 2017.

14.5 Data Processing Scenarios

The Data Processing Scenarios results from the iterations on the prototypes and will be defined with the users and SMEs.

The first pass scenario reads the data repository of articles, extracts articles with Leica Microscope. This is stored on a file for user confirmation. The process iterates until the number of articles reaches an Impact Metrics can be defined.
The second and subsequent scenario is defined by the results of the first pass scenario.

14.6 Limitations

This is a first prototype of a TDM Assisted Research Impact using Leica Microscope. The results are limited by available data, available SMEs, and interface to the OMTD platform. As of this writing, Frontiers is still to confirm an available SME that can confirm the requirements and specifications.

This prototype handles Leica Microscopes only based on articles available in the CORE and OpenAIRE repository. It is intended as a proof-of-concept how the research impact of other resources in research, can be measured.

It will be implemented as a stand-alone Text and Data Mining module and will be integrated into OMTD as the APIs and components become available.

14.7 Release Plan

A first version of the prototype will be available by early October based on the first pass scenario. This will be updated on a regular basis resulting from discussions with the users and SMES and availability of the OMTD platform, interfaces, and components.
15. **SC-E: Rock Art Mining**

This application design document represents Frontiers’ efforts so far and is subject to further updates according to concretization of the application’s requirements.

The objective of this text mining use case is to extract information from articles related to rock art (e.g.: ancient cave paintings) to ease and speed up the research on this topic.

The process will be as follow; identifying articles, downloading, extracting information, storing and publishing the results.

Articles that potentially contain information about rock art will be identified by a list of thematic keywords given by the researchers, then downloaded. As of today, the bibliographic source used is CORE.

- **CORE** (https://core.ac.uk), Core is a repository aggregating open access bibliographic data from over 1100 sources around the world. It currently contains 80 million open access articles of which over 6 million are full text articles. This is very convenient since no special extraction step is necessary.

Once the texts of the articles are downloaded, the following information will be extracted out of them:

- Sites (e.g.: Cave of Altamira)
- Coordinates (e.g.: 43.38, -4.12)
- Dates (e.g.: 12’990 ± 200 BP)
- Dating methods (e.g.: Radiocarbon)

The annotations will be stored in a relational database along with the publication metadata and some computed fields. As requested, they will be made available to the researchers through CSV files, exported on demand. The annotated texts will also be published to an online annotation service for correction and context purposes.

15.1 **Resources**

15.1.1 **Document Formats**

**XMI** (XML Metadata Interchange)

- Reference: http://www.omg.org/spec/XMI/2.5.1
- Intended use: storing annotated documents as text

**JSON** (JavaScript Object Notation)

- Reference: http://www.json.org
- Intended use: storing documents retrieved from bibliographic source
15.1.2 Knowledge Bases
No knowledge base used. Instead, a list of themes and dating methods are provided by the researchers to perform search and the classification of articles.

15.1.3 Tools/Components

**StanfordNamedEntityRecognizer** (as DKPro component)

- Version (DKPro): 1.8.0
- Link to resource:
  - [http://repo1.maven.org/maven2/de/tudarmstadt/ukp/dkpro/core/de.tudarmstadt.ukp.\ dkpro.core.stanfordnlp-gpl/1.8.0/](http://repo1.maven.org/maven2/de/tudarmstadt/ukp/dkpro/core/de.tudarmstadt.ukp.dkpro.core.stanfordnlp-gpl/1.8.0/)
- Link to documentation:
  - [https://dkpro.github.io/dkpro-core/releases/1.8.0/docs/component-reference.html#engine-StanfordNamedEntityRecognizer](https://dkpro.github.io/dkpro-core/releases/1.8.0/docs/component-reference.html#engine-StanfordNamedEntityRecognizer)
- Intended use: annotate locations
- Input/Output: CAS
- Licence: Apache License 2.0

**OpenNlpNamedEntityRecognizer** (as DKPro component)

- Version (DKPro): 1.8.0
- Link to resource:
  - [http://repo1.maven.org/maven2/de/tudarmstadt/ukp/dkpro/core/de.tudarmstadt.ukp.\ dkpro.core.languagetool-asl/1.8.0/](http://repo1.maven.org/maven2/de/tudarmstadt/ukp/dkpro/core/de.tudarmstadt.ukp.dkpro.core.languagetool-asl/1.8.0/)
- Link to documentation:
  - [https://dkpro.github.io/dkpro-core/releases/1.8.0/docs/component-reference.html#engine-OpenNlpNamedEntityRecognizer](https://dkpro.github.io/dkpro-core/releases/1.8.0/docs/component-reference.html#engine-OpenNlpNamedEntityRecognizer)
- Intended use: annotate dates
- Input/Output: CAS
- Licence: Apache License 2.0

**UIMA Fit**

- Version (DKPro): 1.8.0
- Link to resource: [http://repo1.maven.org/maven2/org/apache/uima/uimafit\-core/2.2.0/](http://repo1.maven.org/maven2/org/apache/uima/uimafit-core/2.2.0/)
• Intended use: Analysis engine pipeline  
• Data format: UIMA Analysis engine  
• Licence: Apache License 2.0  

UIMA Ruta  
• Version: 2.6.1  
• Link to resource: http://repo1.maven.org/maven2/org/apache/uima/ruta-core/2.6.1/  
• Link to documentation: https://uima.apache.org/d/ruta-current/tools.ruta.book.html  
• Intended use: Rules Engine to annotate sites and coordinates  
• Data format: CAS, RUTA scripts  
• Licence: Apache License 2.0  

<!--This section is subject to change since we might also use different NERs depending on the results-->  

15.1.4 Services  
WebAnno  
• Address: https://webanno.openminted.eu  
• Performed Tasks: Store annotated articles for corrections purposes  
• Input/Output formats: XMI  
• Hosting institution: OMTD  
• Type of interface (REST/SOAP/Other): Web user interface  

<!--This section is subject to change since WebAnno does not expose a supported API to publish-->  

15.1.5 Other  
CORE  
• Address: https://core.ac.uk/  
• Performed Tasks: source of articles  
• Input formats: Api calls  

15.2 Deployment Plan  
As a first step, the application will run locally using a UIMA Fit pipeline.  
The next step is to adapt the application to integrate with the OMTD platform. OMTD data provider  
and annotation services will be used and the application will be embedded in one or many Docker  
images to be deployed to the platform. We also hope the decoupling will allow us to run the extraction  
over multiple data sources and will make it possible to scale out if necessary.  

<!--This section will be updated as we gain more knowledge and experience of the OMTD platform-->  

15.3 Data Interfaces
CORE

- Protocol: REST API
- Data transfer format: JSON/PDF
- Level of automation: full

Database

- Protocol: SQL
- Data transfer format: Text
- Level of automation: full

WebAnno

- Protocol: HTTP
- Data transfer format: XMI
- Level of automation: none, manual

<This section is subject to change since we want to use the OMTD annotation service>

15.4 User Interfaces

No user interface is currently planned.

15.5 Data Processing Scenarios

1. Searching

   Potential relevant articles are searched using the HTTP API of CORE. The API provides a way of searching using Elasticsearch queries.

   The query is formed using the list of themes and dating methods provided by the researchers.

2. Downloading

   The search query returns a paginated list of publication metadata as JSON. One of the great features is the possibility to get the full text of the publication at the same time, if it is available.

3. Converting

   When the full text of the publication is available, thankfully, we do not have to perform a conversion. However, CORE does not only contain full text articles. Therefore, conversion from PDF to text might be necessary.

   The annotated articles given by researchers to train the NERs were only available as PDF. Therefore, they had to be converted. Some of them even required the use of an OCR (Optical character recognition) because they were containing scans of texts.
4. **Extracting**

   The following valuable information needs be extracted:
   - Date
   - Site
   - Dating methods
   - Coordinates

   Ruta rules and the OpenNLP named entity recognizer are used to annotate the dates.

   Two kinds of annotations are created for the sites. The first one is the result of the Standford NER to annotate all locations. The second set of annotations contains only cave names, which is valuable information for the researchers. Those are the product of rules having a low recall but high precision. Researchers can then use both annotations to identify articles.

   The dating methods and coordinates are identified using Ruta rules.

   <This section is subject to change depending on the performance of the NERs/Rules>

5. **Classifying**

   The above extraction process yields relatively low precision and can return false positives. Therefore, based on the extraction’s output, we determine if the articles are indeed associated with the theme or not.

6. **Storing**

   The annotations are stored along with the metadata of the articles into a relational database.

7. **Exporting**

   Annotations will be exported to CSV files when the researchers request new data.

   The annotated articles are manually exported to WebAnno, so that the researchers can review and give feedback regarding the annotations. They can also use it to locate the annotations in the articles.

   <This section is subject to change since OMTD contains a service to publish annotated articles. Also, depending on the license of some articles, it may be forbidden to publish them under an annotated form.>

15.6 **Limitations**

   The first limitation is the number of articles related to rock art available on CORE. After a primary investigation, it seems there are fewer than a thousand relevant publications.
To store annotations, WebAnno is currently used. Despite its many benefits, it does not support an API to automatically publish annotated texts. This process has to be done manually for each set of processed articles. It does not appear to be a big limitation since the number of relevant articles related to this use case on CORE data source is relatively small.

A third and final limitation is the problem of article license preventing their publication under an annotated form.

15.7 Release Plan

Beta - September 2017

First release using a UIMA Fit pipeline.

Final - November 2017

In this second release, the application will have been adapted to be deployed and take advantages of the OMTD platform.

Also, following user feedback the name entity recognizers might have to be fine-tuned.
16. AS-C Appendices

1. Corpus design
   - Florilège corpus
   - Purpose

The Florilège corpus is a set of extracts of 106 full-text scientific papers manually annotated with the information of the Microbial Biodiversity use case. It is used for the training and evaluation of the AS-C TDM workflow tools.

   - Design
   - Document selection

The references have been provided by Florilège team that is composed of INRA microbiologists from the Food thematic group of the INRA Metaprogramme MEM Meta-omics and microbial ecosystems.

They have selected the references so that they cover the topics of food positive flora.

Document harvest

The URL of the full-papers has been identified using Google Scholar. The html version of the full-papers have been manually downloaded from the publisher web sites. It is worth noting that the Save As command of the web browsers is not usable because the loading and display of full-papers subparts are triggered by the scrollbar.

Document conversion

The html version of the papers has been converted into text format (UTF8 encoding) by applying html to text parsers. Manual curation of the result was needed to achieve a good quality of display for fine-grained manual annotation.

Document annotation

The annotation follows INRA procedure as described in [Chaix et al., 2016], including use of AlvisAE Annotation Editor, double-blind annotation, use of annotation monitor and community
management tools available on a RedMine project. The guidelines document is available at https://docs.google.com/document/d/1VgTrSSh6yo2xLk0QYlbV1JrmnkxFUWWjgHFsG9Cm7nE/edit

The Entity types are

- Taxon
- Habitat
- Molecule
- Phenotype
- Use

They are all described by hierarchical knowledge bases.

The event types are binary or ternary:

- Lives_in : Taxon Lives_in Habitat
- Produces : Taxon Produces Molecule
- Produces_in: Taxon Produces Molecule _in Habitat
- Degrades : Taxon Degrades Molecule
- Degrades_in: Taxon Degrades Molecule _in Habitat
- Exhibits : Taxon Exhibits Phenotype
- Exhibits_in: Taxon Exhibits Phenotype _in Habitat
- Studied_for :Taxon Studied_for Use
- Studied_for_in: Taxon Studied_for Use _in Habitat
- Inhibits_growth: Taxon or Molecule Inhibits_growth Taxon
- Inhibits_growth_in: Taxon or Molecule Inhibits_growth Taxon _in Habitat
- Stimulates_growth_in: Taxon or Molecule Stimulates_growth Taxon_in Habitat
- Corpus composition

The list of references is here.
Format

The corpus with the annotation is distributed in the BioNLP-ST format.

License

The licenses of the papers are various. Only short passages of the papers are fully-annotated. As such passages with the manual annotations are publicly distributed with the CC-BY-SA licence.
● Microbial Biodiversity Food corpus

● Purpose

The purpose of the Microbial Biodiversity Food corpus is to gather all relevant scientific full-papers on positive microbial flora in food.

● Design

● Document selection

The relevant microbial taxa have been first identified by a specialist of systematics. INRA has identified in PubMed database all references that mention microorganisms, using all indexes, including MeSH. Bacteria Biotope pipeline of Alvis Suite has been then applied to this corpus to recognize and categorize the habitats.

The references that mention at least one habitat of the food category has been selected. It yields 1,13 millions references. The food category is the one defined in the version BB’16 of OntoBiotope Ontology.

Microbiologists involved in the Florilege project examine the list of the journals with the reference counts and classified them into three classes:

- highly relevant,
- moderately relevant,
- not needed

Two additional criteria have been considered, legal and technical.

Legal: We retained the journals that grant Open Access or for which INRA pays subscriptions.

Technical: we preferred journals such that

- we can easily harvest full-text documents.
- the html version is easily processed: we already developed parsers or the CSS is close to others that INRA already handled.
Document harvest

We retrieved the PubMed bibliographic metadata for the papers that meet all criteria defined above. To get the URL of the documents we used different strategies depending of the journal publisher platform. (1) We use publisher API. (2) We use CrossRef to get the landing page from the DOI, then we parse the landing page to find the URL of the html version of the document. (3) The URL of the papers of some platforms are composed of a publisher URL plus the DOI. In this case, we used a script to generate them.

- License

The full list of DOI (1,15 millions references) counts 8,670 journals. In the current version of the corpus,

- 32% of the paper have free access license (13% papers only have explicit TDM clauses, in 5% of the journals).
- Only ⅛ of the journals are OA journals (1,098 journaux).
- INRA has right to access to 48% of the full papers including INRA subscription plus OA (551,817 articles).
- 52% of the needed papers are then not legally accessible.

The papers are distributed by 30 different platforms (e.g. SpringerLink, Wiley Online Library, HighWire Press, ScienceDirect).

2. User interface

- Main screen of the Annotation Progress Monitor
### Global progress

<table>
<thead>
<tr>
<th>Annotation</th>
<th>Adjudication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documents</td>
<td>Documents</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Entities</td>
<td>Entities</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Relations</td>
<td>Relations</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Progress by annotator

<table>
<thead>
<tr>
<th>Annotator</th>
<th>Arnaud</th>
<th>Ba</th>
<th>Claire</th>
<th>Estelle</th>
<th>Louise</th>
<th>Philippe</th>
<th>Robert</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>0023</td>
<td>0041</td>
<td>0008</td>
<td>0004</td>
<td>0002</td>
<td>0003</td>
<td>0002</td>
</tr>
<tr>
<td>0106</td>
<td>1107</td>
<td>1010</td>
<td>1010</td>
<td>1010</td>
<td>1010</td>
<td>1010</td>
<td>1010</td>
</tr>
<tr>
<td>Finished</td>
<td>Finished</td>
<td>Finished</td>
<td>Finished</td>
<td>Finished</td>
<td>Finished</td>
<td>Finished</td>
<td>Finished</td>
</tr>
</tbody>
</table>

- **Finished**: the annotator has published all documents. Perseverance pays.
- A bold status means that the `ossesosity.save` property has been injected.
● Mock-up screens of the parent Florilège application
Usages of the biotope and phenotype information

- Semantic search engine
- Genetics comparison
- Data integration
- Visualisation
3. Data processing diagrams

1. Call tree

AS-C Application Design
Ontology/lexicon Analyzer
Lexicon Lemmatization
AS-C Production Processing
Corpus Builder
AS-C Automatic Corpus Annotator
Corpus Converter
AS-C NER and Normalizer
Segmenter-POS-Tagger
# D9.2 – Community Driven Applications Design Report

## Relation Extractor

### Data exporter

### Corpus Manual Annotation

### Relation Extraction Model Learner

## 2. Diagram list

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Segmenter-POS-Tagger</strong></td>
<td>Segmentation (words, sentences) et POS tagging for</td>
</tr>
<tr>
<td><strong>AS-C Automatic Corpus Annotator</strong></td>
<td>Automatic annotation of the corpus: recognition and normalisation of entities and relation extraction</td>
</tr>
<tr>
<td><strong>AS-C Data Exporter</strong></td>
<td>Export annotations with references to relevant KB and KB themselves in the parent application format.</td>
</tr>
<tr>
<td><strong>AS-C Production Processing</strong></td>
<td>Main diagram</td>
</tr>
<tr>
<td><strong>AS-C NER and Normalizer</strong></td>
<td>Automatic annotation: recognition and normalization of entities</td>
</tr>
<tr>
<td><strong>Corpus Builder</strong></td>
<td>Corpus design from bibliographic query to full-text paper harvesting.</td>
</tr>
<tr>
<td><strong>Corpus Manual Annotation</strong></td>
<td>Manual annotation</td>
</tr>
<tr>
<td><strong>Relation Extraction Model Learner</strong></td>
<td>Learning a model for relation extraction.</td>
</tr>
<tr>
<td><strong>Lexicon Lemmatisation</strong></td>
<td>Lexicon lemmatisation and export.</td>
</tr>
<tr>
<td><strong>Relation Extractor</strong></td>
<td>Prediction of relations in a corpus by a learned model.</td>
</tr>
<tr>
<td><strong>Corpus Converter</strong></td>
<td>Corpus Converter converts documents from diverse sources and in various formats into a unified AlvisNLP/ML--OMTD representation</td>
</tr>
<tr>
<td><strong>AS-C Application Design</strong></td>
<td>Definition of the workflow and resource evaluation</td>
</tr>
</tbody>
</table>
3. Diagrams

- **AS-C Application Design**

**AS-C Application Design**

AS-C Application Design represents the scenario of the workflow and resource definition phase. It produces indicators on quantity and quality of annotated data.
Lexicon Lemmatization

Process to tagged a lexicon / vocabulary

Lexicon

Corpus Converter

Word Segmentation (WoSMig)

Segmented lexicon

POS Tagging (TreeTagger or GeniaTagger)

Tagged lexicon

Lexicon extraction (Tabular Export)

Lemmatised lexicon

Detailed subprocess:
• Corpus Converter
Ontology/Lexicon Analyzer

The Ontology/Lexicon Analyzer preprocesses an ontology (or lexicon) so that it can be used to normalize entities with the ToMap method. Preprocessing includes tokenization, POS tagging, lemmatization and term structure analysis.

AS-C Production Processing
AS-C Production Processing

AS-C Production Processing represents the whole process for the AS-C use case. It produces data to be integrated to Florilège.

- Corpus Builder

![Diagram showing the process of AS-C Production Processing]

- Corpus Builder
Corpus Builder

Corpus Builder allows to identify and collect relevant scientific articles. It requires external services as content providers. It produces corpora to feed the text mining workflow.

**Used by:**
- AS-E Production Processing
- AS-D Production Processing
- AS-C Production Processing

---

**AS-C Automatic Corpus Annotator**
AS-C Automatic Corpus Annotator

AS-C Automatic Corpus Annotator presents the whole process of automatically annotating a corpus, including named entity recognition and normalization, and relation extraction.

- Corpus Converter

Corpus

Corpus Converter

Named entity recognition and normalization (AS-C NER and Normalizer)

Annotated corpus

All entities

Relation Model

Relation extractor

Annotated corpus

Relations

Annotated corpus

Entities and relations

Knowledge bases

Detailed subprocess
Corpus Converter

Convert Publisher format to OMTD format, count relevant documents and filter empty documents

Use in the superprocess:

- AS-E Production Processing
- AS-D Automatic Corpus Annotator
- Lexicon Lemmatisation

- AS-C NER and Normalizer
AS-C NER and Normalizer

AS-C NER and Normalizer performs entity recognition and normalization for the AS-C use case. Entities include Microorganisms, Chemical molecules, Habitats, Phenotypes and Uses.

- Corpus
  - NCBI taxonomy
    - Lexicon projection (SimpleProjector)
      - Annotated corpus NCBI taxa
  - Regular expressions
    - Regular expression matching (Regexp)
      - Annotated corpus Normalized Microorganism entities (intermediate)
    - Regular expression matching (PatternMatcher)
      - Annotated corpus Normalized Microorganism entities (final)
  - Segmenter and POS tagger
    - Chemical molecule recognition (ChemSpot, smChem or Argo)
      - Annotated corpus Chemical molecules
    - Geographical name recognition (StanfordNER)
      - Annotated corpus Geographical entities
  - Term extraction (YatesExtractor)
    - Annotated corpus Terms
  - Preprocessed OntoBiotype ontology
    - Analyzed terms
      - Rules
  - Preprocessed Use ontology
    - Analyzed terms
      - Rules
  - Habitat and Phenotype entity normalization (ToMapProjector)
    - Annotated corpus Normalized Habitat and Phenotype entities
  - Use entity normalization (ToMapProjector)
    - Annotated corpus Normalized Use entities
Segmenter-POS-Tagger

Segmenter-POS-Tagger performs word segmentation, sentence segmentation and POS tagging of a corpus.

Relation Extractor
Relation Extractor

Relation Extraction predicts binary relations. It requires an annotated corpus, a learning model and learning parameters. It produces an annotated corpus with the predicted relations.
AS-C Data Exporter

AS-C Data Exporter

AS-C Data and KB export transforms CMTC data (annotated corpus and KB) to Florilège compatible format and transfers them to the final Florilège applications. It requires configuration.

Used by AS-C Production Processing diagram
Corpus Manual Annotation

Corpus manual annotation takes a corpus of document and an annotation schema as input. It produces a corpus with reference annotations that are used by the training tools. It first uses the automatic annotation process to produce a pre-annotation that is then corrected by domain experts.

Relation Extraction Model Learner
Relation Extraction Model Learner

Relation Extraction Model Learner enables the training of a relation extraction model. It requires a training corpora and learning parameters. It produces a relation model. Annotated corpora contains train and dev corpus and is obtained from a manually annotated corpora.
17. Appendices of AS-D

- Corpus design
  - Wheat Corpus

- Purpose

The purpose of the Wheat corpus is to gather all relevant scientific full-papers on bread wheat genetics and phenotypes.

- Design

- Document selection

We defined a relevant query with the Biologists and bioinformaticians involved in Wheat use case (breeders, URG1) to get relevant journals from bibliographic databases (WoS). It is:

\[ TS=((\text{wheat OR aestivum}) \text{ AND gene NOT durum})) \text{ AND LANGUAGE: (English)} \text{ AND DOCUMENT TYPES: (Article)} \]

\[ \text{Indexes=SCI-EXPANDED Timespan=All years} \]

that means papers published in journals, written in english where the title or the abstract or the author keywords or WoS keywordplus contain relevant keywords: gene and “wheat or aestivum not durum” that means bread wheat. It yields 1361 different journals.

The experts examined the list of the journals with the reference counts and classified them into three classes:

- highly relevant,
- moderately relevant,
- not needed

Two additional criteria are considered, legal and technical.

Legal: We retained the journals that grant Open Access or for which INRA pays subscriptions.

Technical: we preferred journals such that

- we can easily harvest full-text documents (URL can be automatically generated or CrossRef landing page is correct, publisher API exist).
- the html version is easily processed: we already developed parsers for this journal or the CSS is close to others that we already handled, given INRA list of parsers.

- Document harvest

We retrieved the bibliographic metadata for the papers that meet all criteria defined above from either Web of Science, or Pubmed. To get the URL of the documents, we used different strategies depending of the publisher platform. (1) Publishers offer API. (2) The URL of the papers of some platforms are composed of a publisher URL plus the DOI. In this case, we used a script to generate them. (3) The other URL were got from CrossRef given the DOI.

- License

In the current version of the corpus, the full list of papers (19 811 references) is published in 1 274 journals.

- 7 466 papers (37,7%) in 1 150 journals (90,3%) are not available (to be double-checked).

- 1 113 papers (5,6%) in 12 journals (0,9%) are OA.

1. INRA suscribes to 77 journals (6%): 7445 papers (37,6%).

- Then 634 papers (3,4%) in 16 journals become available after 6 or 12 months out of INRA subscription. A total of 2 639 papers in 27 journals are free access after 6 or 12 months.

- INRA subscription and ISTEX share 2 663 papers (13,4%) in 11 journals (0,9%). ISTEX provides 245 additional papers (1,2%) in 4 journals (0,3%).

245 papers (1,2%) in two journals (0,3%) have multi license (OA and INRA subscription).
250 papers (1.2%) in 9 journals (0.7%) are published by Frontiers.

The 15 platforms are:

- ACS Publications
- ACSESS Digital Library
- APS Journals
- HighWire Press
- HighWire Press AAAS
- HighWire Press Oxford Journals
- Microbiology Society Research Online
- ISTEX
The SAM Blé corpus is a set of full-text scientific papers manually annotated with the information of the Wheat use case. It is used for the training and evaluation of the AS-D TDM workflow tools.

The corpus has been designed by the SAM Blé project funded by FSOV program. It is described in (Nédellec et al., 2014, MTSR).

The references have been provided by SAM Blé breeder participants. They have selected relevant papers about genetic markers and phenotypes of bread wheat on Springer platform. The restriction to Springer is due to technical limitations of html parsing.

The html version of the full-papers have been manually downloaded from Springer journal web sites.

1. Document conversion

The html version of the papers has been converted into text format (UTF8 encoding) by applying html to text parsers. Manual curation of the result was needed to achieved a good quality of display for fine-grained manual annotation.

The annotation follows INRA procedure, as described in [Nédellec et al, 2014], including the use of AlvisAE Annotation Editor and double-blind annotation. The annotators were trained and
provided with detailed guidelines (see references below). The schema can be found in the guidelines document and in [Nédellec et al, 2014].

Objects:
- Marker
- Gene
- Trait
- Phenotype
- Variety
- Allele Size
- Type

Relations:
- marker_allele_size(Marker, Allele_size)
- marker_of_type(Marker, Type)
- marker_tags_gene(Marker, Gene)
- gene_expressioned_by_phenotype(Gene, Phenotype)
- variety_has_gene(Variety, Gene)
- variety_has_phenotype(Variety, Phenotype)
- trait_has_value(Trait, Phenotype)
- marker_size_in_vriety(Marker, Allele_size, Variety)
- marker_tags_gene_in_vriety(Marker, Gene, Variety)
- gene_has_phenotype_in_vriety(Gene, Phenotype, Variety)
- trait_has_phenotype_in_vriety(Trait, Phenotype, Variety)

- Corpus composition

The list of references is available [here].

- Format

The corpus with the annotations is distributed in the BioNLP-ST format.

- License

The licenses of the papers are various. Only short passages of the papers are fully-annotated. As such, passages with the manual annotations are publicly distributed with the CC-BY-SA licence.

- References


User interfaces

- GnpIS interface

Home page: the user can search using the query box

The result of a search is displayed as follows, including results related to scientific literature. The user can see the list of results relevant to her query and consult them.
Information on publications are also displayed on gene specific records. The user can interact as follows:

- **Select a group of publications and get the link to enriched snippets.**
- **Click on one article title and get the link to the enriched snippet.**
- **Filters according to the entities found in the publications.**
- **Specific objects.**
- **Information on entities (genes, varieties...) annotated in the text are shown in the table.**
- **The publications are sorted according to the number of entities detected along with the gene name from the highest to the smallest.**
In addition to bibliographic information on publications, the user can display snippets of text where the relevant information was identified using the text mining workflow. The user can further refine the results and finally access full text of the articles through a web link.
WheatIS search

Home page: the user can search using the simple query box

Search in all WheatIS nodes

Examples: fhb, wmc430, Triticum, TRAES3RF00100011CFD

Result interface (all data)
The results of the query “FHB” show both data and publication information. The user has selected “fhb” in the keyword search box.

Option 1: The results show the concepts detected in the publication.

Option 2: The results show the title of the article, the publication title, and the concepts detected in the publication.

Links to:
- Option 1: the full text page (publisher page)
- Option 2: the annotated article extractions page

Result interface (publication data only)
In addition to bibliographic information on publications, the user can display snippets of text where the relevant information was identified using the text mining workflow. The user can further refine the results and finally access full text of the articles through a web link.
Data processing diagrams
  - Call Tree

AS-D Application Design
  - Corpus Builder
  - AS-D Automatic Corpus Annotator
    - Corpus Converter
    - AS-D NER and Normalizer
      - Segmenter-POS-Tagger

AS-D Production Processing
  - Corpus Builder
  - AS-D Automatic Corpus Annotator
    - Corpus Converter
    - AS-D NER and Normalizer
      - Segmenter-POS-Tagger
AS-D Data Exporter

Ontology/Lexicon Analyzer

Diagram List

AS-D Application Design: AS-D Application Design represents the scenario of the workflow and resource definition phase. It produces indicators on quantity and quality of annotated data.

AS-D Production Processing: AS-D Production Processing represents the whole process for the AS-D use case in the Production Processing scenario. It produces data to be integrated to GnpIS and WheatIS search.

Corpus Builder: Corpus Builder allows to identify and collect relevant scientific articles. It requires external services as content providers. It produces corpora to feed the text mining workflow.

AS-D Automatic Corpus Annotator: AS-D Automatic Corpus Annotator represents the corpus pre-processing and annotation for the AS-D use case. It produces a corpus enriched with entities.

Corpus Converter: Convert Publisher format to OMTD format, count relevant documents and filter empty documents.

AS-D NER and Normalizer: AS-D NER and Normalizer performs entity recognition and normalization for the AS-D use case. Entities include Taxa, Genes, Markers and Phenotypes.

Segmenter-POS-Tagger: Segmenter-POS-Tagger performs tokenization, sentence segmentation and POS tagging of a corpus.

Ontology/Lexicon Analyzer: the Ontology/Lexicon Analyzer preprocesses an ontology (or lexicon) so that it can be used to normalize entities with the ToMap method. Preprocessing includes tokenization, POS tagging, lemmatization and term structure analysis.

AS-D Data Exporter: AS-D Exporter transforms OMT data (annotated corpus and KB) to URGl compatible format and transfers them to the final URGl applications. It requires configuration.

Diagrams

AS-D Application Design
AS-D Application Design

AS-D Application Design represents the scenario of the workflow and resource definition phase. It produces indicators on quantity and quality of annotated data.

AS-D Production Processing
AS-D Production Processing

AS-D Production Processing represents the whole process for the AS-D use case in the Production Processing scenario. It produces data to be integrated to GnipS and Wheat Data Search.

- Corpus Builder
  - Wheat Corpus
  - AS-D Automatic Corpus Annotator
  - Annotated corpus (All entities)
  - AS-D Data Exporter
  - Enriched URG1 DB
  - Reference lists for URG1 search engines

Caption:
- Service
- Detailed process
- Corpus
- Condition
- External KB
- Internal Knowledge Base
- Last process (Tools not more detailed)
Corpus Builder

Corpus Builder allows to identify and collect relevant scientific articles. It requires external services as content providers. It produces corpora to feed the text mining workflow.

Used by:
- AS-E Production Processing
- AS-D Production Processing
- AS-C Production Processing

---

AS-D Automatic Corpus Annotator
AS-D Automatic Corpus Annotator

AS-D Automatic Corpus Annotator represents the corpus pre-processing and annotation for the AS-D use case. It produces a corpus enriched with entities.

AS-D Production Processing
Corpus Converter

Convert Publisher format to OMTD format, count relevant documents and filter empty documents

Use in the superprocess:

- AS-F Production Processing
- AS-D Automatic Corpus Annotator
- Lexicon Lemmatisation

- AS-D NER and Normalizer
AS-D NER and Normalizer

AS-D NER and Normalizer performs entity recognition and normalization for the AS-D use case. Entities include Taxa, Genes, Markers and Phenotypes.

AS-D Automatic Corpus Annotation

Text Wheat Corpus

- NCBI taxonomy
- Gene lexicon
- Marker lexicon
- Regular expressions

Lexicon projection (SimpleProjector)

- Regular expression matching (Regexp)

Segmenter and POS tagger

- Regular expression matching (PatternMatcher)

Term extraction (YesiaExtractor)

Phenotype entity normalization (ToMapProjector)

- Preprocessed WheatPhenotype ontology
  - Analyzed terms
  - Rules

- Annotated corpus
  - Normalized Taxon
  - Gene and Marker entities

- Annotated corpus
  - Normalized Taxon
  - Gene and Marker entities (final)

- Annotated corpus
  - Terms

- Annotated corpus
  - Normalized Phenotype entities

- Annotated corpus
  - All entities

Segmenter-POS-Tagger
Segmenter-POS-Tagger

Segmenter-POS-Tagger performs word segmentation, sentence segmentation and POS tagging of a corpus.

- Corpus
  - Latin expression lexicon
    - Lexicon projection (SimpleProjector2) → Annotated corpus Latin expressions
  - Regular expressions
    - Regular expression matcher (Regexp) → Annotated corpus Numbers and dates
    - Word segmenter (WoSMig) → Annotated corpus Words (intermediate)
      - Preprocessing for hyphens (PatternMatcher, WoSMig) → Annotated corpus Words (final)
      - Sentence segmenter (SeSMig) → Annotated corpus Sentence
        - POS tagger (Treetagger or GentaTagger) → Annotated corpus POS, lemmas
          - Annotated corpus Words, sentence, POS, lemmas

- Ontology/Lexicon Analyzer
Ontology/Lexicon Analyzer

The Ontology/Lexicon Analyzer preprocesses an ontology (or lexicon) so that it can be used to normalize entities with the ToMap method. Preprocessing includes tokenization, POS tagging, lemmatization and term structure analysis.

AS-D Data Exporter

AS-D Exporter transforms OMT data (annotated corpus and KB) to URG1 compatible format and transfers them to the final URG1 applications. It requires configuration.

AS-D Production Processing
18. **Appendices of AS- E**

- **Corpus design**
  - Plant Development corpus (PlantDev)
  - Purpose

The purpose of the Plant Development corpus is to gather all relevant scientific full-papers on the model plant *Arabidopsis thaliana* development.

- **Design**
- **Document selection**
- **Document selection**

We defined a relevant query with the plant biologists involved in SeeDev use case (IJPB researchers) to get relevant journals from bibliographic databases (WoS). It is:

\[
\text{TS} = (\text{Arabidopsis AND Seed AND gene* AND (development OR maturation OR metabolism)}) \text{ AND LANGUAGE: (English)} \text{ AND Indexes=SCI-EXPANDED Timespan=1996-2016}
\]

that means papers published in journals, written in english where the title or the abstract or the author keywords or WoS keywordplus contain relevant keywords. It yields 5 480 papers in 432 different journals.

4305 in PubMed: Search (((Arabidopsis AND Seed AND gene* AND (development OR maturation OR metabolism)) AND full text[sb]) AND english[Language] Filters: Full text

The experts examines the list of the journals with the reference counts and classified them into three classes:

- highly relevant,
- moderately relevant,
- not needed
Two additional criteria are considered, legal and technical.

**Legal:** We retained the journals that grant Open Access or for which INRA pays subscriptions.

**Technical:** we preferred journals such that

- we can easily harvest full-text documents (URL can be automatically generated or CrossRef landing page is correct, publisher API exist).
- the html version is easily processed: we already developed parsers for this journal or the CSS is close to others that we already handled, given INRA list of parsers.

**Document harvest**

We retrieved the WoS bibliographic metadata for the papers that meet all criteria defined above. To get the URL of the documents we used different strategies depending of the journal publisher platform.

1. We use publisher API.
2. We use CrossRef to get the landing page from the DOI, then we parse the landing page to find the URL of the html version of the document.
3. The URL of the papers of some platforms are composed of a publisher URL plus the DOI. In this case, we used a script to generate them.

- **License**
  - Only 837 papers are Open Access (18%).
  - According to PubMed, 66% are free access.

The papers are distributed by 30 different platforms (e.g. SpringerLink, Wiley Online Library, HighWire Press, ScienceDirect).
• **User interfaces**
  - FLAGdb++ interface

Data access for end-users of the final application

→ Home screen of FLAGdb++ application
Data access from a simple keyword search

First access to data from information extraction via the keyword search (from the Gene Classification menu)
○ From keyword search to feedback interface

○ Data access from gene locus and relation search
From relation search to feedback interface
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. AGl15 could bind DNA as a homodimer in vitro. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
• Data processing diagrams
  - Call tree

8. **AS-E Production Processing**

6. **Corpus Builder**
   [https://www.draw.io/?state=%7B%22ids%22:%5B%220Bwefobw4P7poYzAtS0xQOEICQzA%22%5D,%22action%22:%22open%22,%22userId%22:%222101729640835482598083%22%7D-0Bwefobw4P7poYzAtS0xQOEICQzA](https://www.draw.io/?state=%7B%22ids%22:%5B%220Bwefobw4P7poYzAtS0xQOEICQzA%22%5D,%22action%22:%22open%22,%22userId%22:%222101729640835482598083%22%7D-0Bwefobw4P7poYzAtS0xQOEICQzA)

7. **Corpus Converter**
   [https://drive.google.com/file/d/0B5W_0bNv7zKQbIBKTGdGcVAxNTQ/view?usp=sharing](https://drive.google.com/file/d/0B5W_0bNv7zKQbIBKTGdGcVAxNTQ/view?usp=sharing)

8. **AS-E Automatic Corpus Annotator**

8.1 **AS-E NER**

8.1.1 **AS-E NER Preprocessor**
  8.1.1.1 **AS-E Protein and Gene disambiguation**

8.1.2 **AS-E Entity Recognizer**
  8.1.2.1 **CRF Predictor**

8.1.3 **AS-E Entity Normalizer**
   [https://www.draw.io/?state=0B9ehTDy8eYfnWlVirTm1TMDFlaDA](https://www.draw.io/?state=0B9ehTDy8eYfnWlVirTm1TMDFlaDA)

8.2 **Relation extractor**

9. Resource Update:
   a. **Ontology/Lexicon Analyzer**
   b. **Lexicon Lemmatization**
      i. **Corpus Converter**
   c. **Relation Extraction Model Learner**
   d. **Corpus Builder**
   e. **CRF Model Learner**

10. **AS-E Application Design**
- **Diagram list**

**AS-E Production Processing** : Main SeeDev diagram - Production Processing

**Corpus Builder** : Corpus design from bibliographic query to full-text paper harvesting. [https://www.draw.io/?state=%7B%22ids%22:%5B%220Bwefobw4P7poYzAtS0xQOEICQzA22%5D,%22action%22:%22open%22,%22userId%22:%221210729640835482598083%22%7D - G0Bwefobw4P7poYzAtS0xQOEICQzA](https://www.draw.io/)

**Corpus Converter** : Corpus Converter converts documents from diverse sources and in various formats into a unified AlvisNLP/ML-OMTD representation [https://drive.google.com/file/d/0B5W_0bNv7zKQblBKTGdGcVAXNTQ/view?usp=sharing](https://drive.google.com/file/d/0B5W_0bNv7zKQblBKTGdGcVAXNTQ/view?usp=sharing)

**AS-E Automatic Corpus Annotator** : Automatic annotation of corpus for AS-E

**AS-E NER** : Entity extraction for AS-E

**AS-E NER Preprocessor** : Fixed entity recognition for AS-E (SeeDev)

**AS-E Protein and Gene disambiguation** : Protein and gene recognition from typographic information (html)

**Lexicon Lemmatization** : Lexicon lemmatization and export.

**AS-E Entity Recognizer** : Entity recognition by term projection, pattern application and learning (general)

**CRF Predictor** : Entity prediction with a CRF model (general)

**AS-E Entity Normalizer** : Entity normalization with BioYaTeA and ToMap (general) [https://www.draw.io/-G0B9ehTDy8eYfnWIVrTm1TMDFIaDA](https://www.draw.io/-G0B9ehTDy8eYfnWIVrTm1TMDFIaDA)

**Relation extractor** : Prediction of relations in a corpus by a learned model.

**Ontology/Lexicon Analyzer** : Lexicon and ontology preprocessing for ToMap entity normalization (general)

**Lexicon Lemmatization** : Lexicon lemmatization and export.

**Relation Extraction Model Learner** : Learning a model for relation extraction.

**CRF Model Learner** : CRF model learning for entity recognition (general)

**AS-E Application Design** : Definition of the workflow and resource evaluation
- Diagrams

- AS-E Production Processing

**AS-E Production Processing:**
Main TM process for SeeDev use case

Subprocess detailed:
- Corpus Builder
- Corpus Converter
- AS-E Automatic Corpus Annotator

---

**Corpus Builder**

- Corpus and metadata
- Corpus and OMTD metadata
- Annotated corpus and metadata
- Curated corpus

**Corpus Converter**

**AS-E Automatic Corpus Annotator**

**Curation Merger**

**Data exporter (Annotation export, PlantDevKB export)**

**Data for FlagDB++ application**

**Curated Data**

**Data curation**
Corpus Builder

Corpus Builder allows to identify and collect relevant scientific articles. It requires external services as content providers. It produces corpora to feed the text mining workflow.

Used by:
- AS-E Production Processing
- AS-D Production Processing
- AS-C Production Processing

[Diagram of the Corpus Builder process]

- Corpus Converter
Corpus Converter

Convert Publisher format to OMTD format, count relevant documents and filter empty documents

Use in the superprocess:
- AS-E Production Processing
- AS-D Automatic Corpus Annotator
- Lexicon Lemmatisation

- AS-E Automatic Corpus Annotator
AS-E Automatic Corpus Annotator:
Main TM process for SeeDev use case

Super processes:
- AS-E Production Processing

Detailed sub process:
- AS-E NER
- Relation extractor

- AS-E NER
AS-E NER : Named entity recognition process for AS-E Use cases

Use in the superprocess :
- AS-E Automatic Corpus Annotator

Sub process detailed :
- AS-E NER Preprocessor
- AS-E Entity recognizer
- AS-E Entity normalizer

- AS-E NER Preprocessor
AS-E NER Preprocessor: Fixed entity annotation

Use in the superprocess:
- AS-E NER

Sub process detailed:
- AS-E Protein and gene disambiguation

AS-E Protein and Gene disambiguation
AS-E Protein and Gene disambiguation:
Process to choose if an entity is a gene or a protein
Use in the superprocess:
- AS-E NER Preprocessor

AS-E Entity Recognizer
AS-E Entity Recognizer: Process used for entity recognition

Use in the super-process: 
As-E NER

Sub-process detailed:
- CRF Predictor

- CRF Predictor
CRF Predictor

The CRF Predictor uses a trained CRF model to predict entities in a corpus. It relies on the Wapiti toolkit.

- AS-E Entity Normalizer
AS-E Entity Normalizer: detection and categorisation with ontologies

Use in the superprocess: AS-E NER

- Annotated Corpus Entities
- Term extraction (YateExtractor)
- Annotated corpus Terms
- Preprocessed ontologies
- Analyzed terms
- Rules
- Entity normalization (ToMepProjector)
- For each entity type
- Annotated Corpus Normalized Entities

- Relation Extractor
Relation Extractor

Relation Extraction predicts binary relations. It requires an annotated corpus, a learning model and learning parameters. It produces an annotated corpus with the predicted relations.

- Lexicon Lemmatization
Lexicon Lemmatization

Process to tagged a lexicon / vocabulary

- Ontology/Lexicon Analyzer

```
Lexicon
     ↓
Corpus Converter
     ↓
Word Segmentation (WoSMig)
     ↓
Segmented lexicon
     ↓
POS Tagging (TreeTagger or GeniaTagger)
     ↓
Tagged lexicon
     ↓
Lexicon extraction (Tabular Export)
     ↓
Lemmatised lexicon

Language model
```

Detailed subprocess:
- Corpus Converter
Ontology/Lexicon Analyzer

The Ontology/Lexicon Analyzer preprocesses an ontology (or lexicon) so that it can be used to normalize entities with the ToMap method. Preprocessing includes tokenization, POS tagging, lemmatization and term structure analysis.

- Relation Extraction Model Learner
Relation Extraction Model Learner

Relation Extraction Model Learner enables to train a relation extraction model. It requires a training corpora and learning parameters. It produces a relation model. Annotated corpora contains train and dev corpus and is obtained from a manually annotated corpora.

- CRF Model Learner

CRF Model Learner

The CRF Model Learner trains a CRF model using annotations from a training corpus and the Wapiii toolkit. The created model will be used in subsequent tasks to tag entities.
AS-E Application Design

Use-cases: AS-E

AS-E Application Design represents the scenario of the workflow and resource definition phase. It produces indicators on quantity and quality of annotated data.