Building a French-speaking community around UIMA, gathering research, education and industrial partners, mainly in Natural Language Processing and Speech Recognizing domains

Nicolas Hernandez, Fabien Poulard, Matthieu Vernier, Jérôme Rocheteau

LINA (CNRS - UMR 6241) – University of Nantes
2 rue de la Houssinière – B.P. 92208, 44322 NANTES Cedex 3, France
first.last@univ-nantes.fr

Abstract

We report on the efforts the UN-LINA has made to build a UIMA French-speaking community both in Natural Language Processing and Speech Recognizing domains that would bring together researchers, industrials and educational interests. The intentions of building this community are twofold: to encourage the French-speaking academic and industrial organizations which have not yet adopt a middleware solution to use UIMA as a common development framework and middleware architecture for their research and engineering projects; and to improve the collaborative development of common UIMA-based NLP tools and components for processing French. We present the services we have set up as well as the resources we distribute freely under open licences to accomplish this objective. Most of them are currently available on the `uima-fr.org` Web Portal. They consist of: A web portal to discuss and exchange information about UIMA; A bundle of scripts and resources for automatically installing the whole of the Apache UIMA SDK; A bundle of UIMA-based components including some French NLP preprocessing components, a type mapper and a semantic rule-based analyser; A bundle of UIMA tools including an Analysis Engine Apache Maven archetype and an advanced web rest server; Course and training materials.

1. Introduction

Nowadays, it is crucial for a NLP (Natural Language Processing) laboratory which aims at playing a role within the national and the international community to acquire a robust middleware backbone to support its research and engineering activities. The issues are numerous:

- to ensure interoperability among the team members and with the project partners;
- to reuse existing software tools and consequently to benefit from preceding development efforts;
- to go beyond the prototype stage and to make possible the transfer toward industrial releases; In particular by taking into account scalable issues and distributed architecture;
- to be able to build more complex business applications;
- to be promptly operational and responsive to answer scientific and engineering national and international project calls;
- to demonstrate its know-how by deploying its software results as web services for example;
- to extend its business activities to data processing other than text such as audio or video.

In France, although GATE[1], NLTK[2] and Nooj[3] have been used as educational and research tools, there is no common agreement on the use of a particular platform for these purposes. Instead, it is not rare than researchers produce ad hoc solutions for their workflow management and language engineering problems.

Since December 2007, the NLP team of the LINA lab. at the University of Nantes (Nantes Atlantic Computer Science Laboratory) has explored the Apache UIMA framework as a middleware architecture to support its educational, research and engineering projects.

By comparison with the above-cited platforms, the Apache UIMA (Unstructured Information Management Architecture[4]) framework (Ferrucci and Lally, 2004) suits our needs for several reasons: Apache UIMA is an open, industrial-strength, scalable and extensible platform for creating, integrating and deploying unstructured (or semi-structured) information (text, audio, video...) management applications which help to build the bridge from unstructured information to structured knowledge. Apache UIMA dissociates the engineering middleware problems from NLP issues, its principles (semantic search and content analytics) result from a standardization effort at OASIS[5], its international community is very active, its Apache license fits research objectives and allows collaboration with industrial partners, and it is integrated in the Eclipse IDE (Integrated Development Environment).

So far the French-speaking community has paid little attention to this framework, mainly because, since Apache UIMA is a recent framework (the first Apache release is from December 2007), there were few NLP components compared to other frameworks like GATE; In particular for processing French. It is also correct to say that although several tools were available from the first release (such as a standalone workflow manager, an annotation viewer or a web REST service UIMA workflow deployer), the graphic user interfaces of these tools have remained quite basic.

[1] `gate.ac.uk`
[2] `www.nltk.org`
[3] `www.nooj4nlp.net`
[4] `incubator.apache.org/uima`
[5] `www.oasis-open.org/committees/uima`

In this paper, we report on the efforts we have made to build a UIMA French-speaking community both in Natural Language Processing and Speech Recognizing domains that would bring together researchers, industrials and educational interests. Our intentions of building this community are twofold:

1. to encourage the French-speaking academic and industrial organizations which have not yet adopt a middleware solution to use UIMA as a common development framework and middleware architecture for their research and engineering projects;

2. to improve the collaborative development of common UIMA-based NLP tools and components for processing French.

We present the services we have set up as well as the resources we distribute freely under open licences to accomplish this objective. Most of them are currently available on the `uima-fr.org` Web Portal. They consist of:

- A web portal to discuss and exchange information about UIMA;
- A bundle of scripts and resources for automatically installing the whole of the Apache UIMA SDK;
- A bundle of UIMA-based components including some French NLP preprocessing components, a type mapper and a semantic rule-based analyser;
- A bundle of UIMA tools including an Analysis Engine Apache Maven archetype and an advanced web rest server;
- Course and training materials.

Among similar efforts all around the world (Germany, Japan, UK, USA), we count the LTI repository at the Carnegie Mellon University[6], the Apache repository[7], the DKPro repository at the Darmstadt University [8] (Gurevych et al., 2007), the Julie lab repository[9] at the Jena University(Hahn et al., 2007) and the U-Compare project repository[10] (Kano et al., 2009).
These efforts are mainly dedicated to host UIMA tools and components. In comparison, we also focus on services and resources to help colleagues to quickly be productive with UIMA. Nevertheless it is important to notice that the project aims at encouraging the creation of a French-speaking community but it is not strictly dedicated to process French.
This project has been supported by both an IBM Unstructured Information Analytics 2008 Innovation Award and several LINA's research projects. The LINA have been using actively the UIMA framework in several ANR (National Research Agency) and local research projects. In the PIITHIE[11] project, we develop and deploy semantic and discourse analyzers as web services, in order to detect plagiarism and text reuse. In the Blogoscopie[12] project, we develop a component for opinionmining in blogs. In the C-Mantic[13] project, we aim at developing a semantic search engine with UIMA for the semantic analysis parts. In the Miles[14] project, UIMA will be used as the architecture to connect various geographically distributed components for speaker recognizing in text transcription. All these projects involve various academic and industrial partners.

2. The UIMA concepts

In the UIMA jargon, the *Common Analysis Structure* (CAS) is the data structure which is exchanged between the UIMA workflow components. It includes the data, subject of analysis and called the *Artefact*, and the metadata, in general simply called the *Annotations*, which describe the data. The annotations are stored in an index within the CAS. The annotations structure is called the *Types System* (TS) and consist of an implementation of a given annotation scheme. A UIMA workflow is made of three types of components: the *Collection Reader* (CR) which imports the data to process (for example from the Web or from the file system...) and turns it into a CAS. The *Analysis Engines* (AE) which literally process the data (including but not restricted to NLP analysis tasks); The annotations result from AE processing. And lastly the *CAS Consumer* (CC) which exports the annotations (for example to a database or to an XML representation of the analysis results).

3. The `uima-fr.org` web portal services

As part of the efforts, we count the launch of a French-speaking web portal about UIMA, `uima-fr.org`. This portal aims at developing a UIMA French-speaking community by providing services for French-speaking users and developers, researchers or professionals from both academic and industrial organizations to discuss and exchange information about UIMA.
Currently, the portal offers three main services allowing anyone to inform and to share informations about UIMA.

- a discussion list `sympa.univ-nantes.fr/wws/info/discussion-uima-fr`;
- a feed aggregator designed to collect posts from the blogs of any members of the community and display them on a single page `uima-fr.org/planet`;
- and a resource repository available under open license;

The first two services, the discussion list and the feed aggregator, aim at collecting FAQ explanations and HOWTO procedures in French. They act as a first step toward a more structured version of the content that a wiki could offer for

[6] `uima.lti.cs.cmu.edu`
[7] `incubator.apache.org/uima`
[8] `www.ukp.tu-darmstadt.de/software/dkpro`
[9] `www.julielab.de/Resources/Software/NLP_Tools.html`
[10] `u-compare.org/components`
[11] `www.piithie.com` financed by the ANR under the Software Technologies Program 2006–2008
[12] `www.blogoscopie.org` financed by the ANR under the Software Technologies Program 2006–2008
[13] `www.c-mantic.org` financed by the ANR under the Data mining and knowledge 2007–2009
[14] Regional project, "Pays de Loire" 2007–2009

example. Topics cover both users and developers interests, dealing with install, use, teaching and development issues both with the Apache UIMA framework and the third-part tools and components.
The third service aims at freely distributing the documentation, the tutorial and the education resources as well as the ready-to-use UIMA-based components and UIMA tools we created.

4. Scripts and resources for installing the Apache UIMA SDK

The Apache UIMA Software development kit (SDK) is made of several tools and dependencies namely the Java Sun Development Kit (JDK), the Eclipse Integrated Development Environment (IDE), some Eclipse plugins and the Apache UIMA framework itself. This basic environment can be extended to include Apache Tomcat or other third-part tools or uima-based components.
Despite the fact that it exists a well-made documentation to help the installation and the use of all these tools and dependencies, it is not always easy to get into it because each tool has it own installation instructions, because it also may require a few engineer skills or sensitivity, because it can take some times to handle all of that...
To avoid all these disheartening aspects, we decided to provide some scripts to assist the download, the configuration, the installation of the UIMA SDK as well as its run within the Eclipse IDE.
We worked out that Eclipse would support the use of UIMA workflows and the development of UIMA-based components. The scripts were dedicated to run on Debian-like systems. They were validated on Ubuntu 8.10 Hardy and 9.04 Jaunty. Currently two versions of them are distributed:

- a light version which requires the launch of a download script to retrieve the tools and the dependencies of the UIMA SDK;
- and a standalone version which directly includes all the required tools as a resources package. The 20100207 version integrated the JDK 6u17, Eclipse Galileo 3.5 IDE, the subeclipse Subversion and the m2eclipse Maven eclipse plugins, the Apache UIMA 2.3.0-incubating framework, the Apache Tomcat 6.0.20 web server and the OpenNLP v1.3 toolkit.

In order to follow the progress of the resources, the scripts were written to work with a property file where the version and the url of the tools to use can be set up. These scripts are distributed under GPLv3 license.

5. UIMA-based components

We present below some French NLP preprocessing components as well as a type mapper and a semantic rule-based components.

5.1. French NLP preprocessing components and type system

In order to open the French NLP community to UIMA, we focused on the development and the distribution of preprocessing components which are commonly used in most of the NLP applications. The underlying idea was to offer a base from which collegues could directly start to work on their own applications and issues without losing times.
The components we worked out and distribute now permit the following processing: URI-based data import, MIME type recognition, text extraction, language recognition, NLP preprocessing (tokenization, stemming, POS tagging, lemmatization). We decided to wrap widely known and used tools whenever it was possible at least for three main reasons: First, it is a tremendous work to redevelop from scratch and we did not have the time neither the fund to do it. Second, so that novice people in UIMA but not in NLP wouldn't be too lost. And last but not the least, in order to enjoy the evolutions of other tools progressing independently.
Namely, we wrap the following tools and libraries: Apache Tika[15] (toolkit for detecting and extracting metadata and structured text content from various documents using existing parser libraries), nGRAMj (a Java based library providing robust and state of the art language recognition/guessing) [16], the Snowball library[17] (French stemmer), Brill (POS tagger) [18], TreeTagger Schmidt (POS tagger and lemmatizer) [19], Flemm (lemmatizer) [20]. In future work, we will include wrappers for TreeTagger and Minipar chunkers.
Our Tika Annotator acts in a complementary way with the Apache one since it works with URI as input and provided Dublin Core compliant markup annotations whenever the types of information are available. Compared to the TreeTagger wrapper of DKPro Repository, we map the TreeTagger outputs to the MulText annotation scheme.
Based on the capabilities of the available preprocessing tools for lemmatization, postagging and chunking in French, we have designed a Multext[21] compliant type system to annotate morphosyntaxic informations, with a particular attention to French language. This type system is currently experimented in our projects. Similarly to the Julie lab and U-compare project's type systems (Hahn et al., 2007; Kano et al., 2009), we intend to make the type system as generic as possible. Our type system is compliant with them as well as offering a specialization for French language.

5.2. A Type Mapper Component

One of the major issue dealing with any workflow management frameworks is the components interoperability. UIMA components only exchange data. So the data structure of the shared data is important since it ensures the interoperability.
Currently, there are at least three proposals for a tool- and

[15] `incubator.apache.org/tika/`
[16] `ngramj.sourceforge.net`
[17] `snowball.tartarus.org`
[18] `en.wikipedia.org/wiki/Brill_tagger`
[19] `www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger`
[20] `www.univ-nancy2.fr/pers/namer/Telecharger_Flemm.htm`
[21] `aune.lpl.univ-aix.fr/projects/MULTEXT` and `nl.ijs.si/ME/` Multilingual POS Tagset projects

domain-independent type system: The CCP meta model type system (Verspoor et al., 2009), the Julie lab's type system (Hahn et al., 2007) and the U-Compare project's type system (Kano et al., 2009). The former consists of a simple annotation hierarchy where the domain semantics is captured through pointers into external resources. The second and the third roughly consist of an abstract hierarchy of NLP concepts covering the various linguistic analysis levels.

According to us, tool and domain-independent type systems should be used whenever it is possible, at least as an example frame. But in our opinion, it will always be necessary to develop software converters from/to the proposed standard solution at least for two main raisons: First, in order to ensure the compatibility with pre-existing annotated data and processing softwares (which may have taken considerable time and funding to develop); this directly concerns the current tool- and domain-independent type systems[22]. Second, the need for new ad hoc type system will always exist for some problem adequacy reasons (unexplored application/domain/language, new or opposite theoretical approach, language dependency) and economic specifications (software resources to use imperatively).

For all these reasons, we decided to develop a generic converter Analysis Engine, called the Type Mapper, to permit the mapping of types or features values into others. The precondition for type mapping is that all the concerned types inherit from *uima.tcas.Annotation* so that they could have a begin and end offset. For each annotation of a given type, our Type Mapper component creates one or several annotations at the same offset. The code is totally generic except it requires both to put in the build path the various type systems classes involved as well as to set up the input/output capabilities of the converter. The mapping rules are specified in an external file as a parameter.

The preprocessing components package described in Section 5.1. includes a version for mapping the POS taggers and lemmatizer outputs to the Multext scheme.

5.3. A semantic rule-based analyser

Analysis is one of the major NLP tasks. A semantic rule-based analyser should enable to create or update annotations according to rules expressed over other annotations.

Currently, the Apache Regular Expression Annotator (RegexAnnotator) is an analysis engine which can create or update annotations but unfortunately it is dedicated to text surface analysis at the character level. The Apache Lucene CAS indexer (Lucas) are CAS consumers that stores annotations in an external index. It is then possible to make some structured searches in a Lucene application. The IBM Semantic Search, also referenced in the Apache UIMA repository, works similarly. These solutions are external applications and are not part of a UIMA workflow (at least natively). The Apache Configurable Feature Extractor (CFE) enables feature extraction from a CAS according to rules expressed using the Feature Extraction Specification Language (FESL). This is assurely a step toward the solution we are looking for but the CFE is a Cas Consumer and it does not enable natively annotation creation or update. The TextMarker component (Kluegl et al., 2009)[23] is a very appealing project but so far it remains very complex to use and quite dependent of the Eclipse environment.

Based on the Type Mapper we presented in Section 5.2., we started to develop a semantic rule-based analysis engine. We start from the Type Mapper since a semantic rule can be considered as a mapping operation from a contextually constrained source type. Currently, we are maintaining two development branches.

The former is based on the Apache UIMA API (Application Programming Interface). From a formal language we defined with ANTLR (ANother Tool for Language Recognition)[24], constraints are dynamically generated over the annotations, then the annotations are filtered according to the constraints. So far, the constraint language permits to specify annotations features and covered text values.

The latter branche follows an alternative approach. The idea was to transpose the problem to another domain where a request language over structured data and its processing engine are already available. The Lucas Annotator and the IBM Semantic Search developers chose to transpose to a database request problem. We chose to transpose the CAS analysis problem to a XML analysis problem. The XPath language offers to express constraints over a structure somehow quite similar to the text structure with the possibility to specify directions within. Futhermore it has several functions, in particular String functions. The major drawback is that XML is by definition a tree structure but not necessary the CAS. We solve the problem by using the JXPath[25] library which applies XPath expressions to graphs of objects of all kinds by setting a context node. So far, the two branches progress at the same level. This work is hosted at the Google forge[26].

6. UIMA tools

Below we present some tools we have been developed during our projects and we wish to distribute to the community.

6.1. An Advanced Web Rest Server

Among the tools available in the Apache repository, the Simple Server permits to provide UIMA analysis as a REST service. The server processes text raw data attached in HTTP request and outputs analysis results in an XML ad hoc format.

For the need of the PIITHIE project, we extended this version in three ways: First, the input source does not need to be attached to the HTTP request but can be specified by an URI (Universal Reference Identifier). The Server automatically upload the resource from the URI thanks to the JAVA API. Second, URI can refer to any resource formats. We included the Tika library in the server side. Third, the

[22] The U-Compare project comes with some ad hoc Type System converters from CCP, OpenNLP and Apache which turn them into the U-Compare Type System. It offers also some U-Compare Type System to OpenNLP.

[23] `tmwiki.informatik.uni-wuerzburg.de`

[24] `www.antlr.org`

[25] `commons.apache.org/jxpath`

[26] `code.google.com/p/uima-type-mapper`

server can process XMI data in input and provides XMI in output. The server can turn the XMI into a CAS as long as the described annotations belong to a type system available in its classpath.
Recently, we decided to distribute this work and to extend it with new features such as PEAR (Processing Engine ARchive), collection and access rights management. This project is hosted at the Google forge[27].

6.2. An Analysis Engine Maven Archetype

Apache Maven[28] is a build manager for Java projects. One of its features is the *archetype* facility which offers a way to define template project.
We built an Analysis Engine Maven Archetype in order to save the best practices we defined as well as to enable new developers jumping board as quickly as possible.
The archetype creates a repository project dedicated to the development of an Analysis Engine adding the UIMA nature to it. It comes with an annotator code and descriptor templates which include generic parameters. These parameters are meant to specify the input/output views/types/encoding. The archetype also creates some basic files such as a README and a LICENSE files.

7. Course and training materials

In order to animate and to train the community to the UIMA framework, we organized a training session during the 10th edition of the LSM/RMLL[29] (Libre Software Meeting) conferences 2009. This session presented how to build and to carry out processing chains and to develop his own UIMA components. For this purpose, we built tutorial-handouts, exercices and answers codes, and videos. We have also used Apache UIMA as a framework for educational purpose. We wrote course materials for Master's programs. We focused on writing UIMA components and interfacing UIMA with WEKA (Machine Leaning Library). All these resources are referenced on the `uima-fr.org` web portal.

8. Conclusion and future works

Many of the mentioned resources are currently dispatched on several web pages (LINA web pages[30], `uima-fr.org` planet and repository, Google forge projects). We are currently setting up a main index in the `uima-fr.org` repository. The UIMA-based components and UIMA tools are distributed under Apache 2.0[31] or GPLv3 [32] license. The training resources are distributed under a double License CC-by-sa fr 2.0[33] and GNU FDL[34]. One of our future works will concern the distribution of the components across an Apache Maven repository since it handles automatically the download of package dependencies.
The list of components we presented remains uncomplete. Indeed, our components bundle includes a Command Line analysis engine which performs on a given view the shell command specified and get the result in a dedicated annotation. This annotator is useful to easily and quickly integrate external softwares. The bundle includes also a XML-to-CAS analysis engine which parses any well formed XML data files and maps the XML structure, the elements and the attributs into generic annotation feature structures.
Some other components and tools are also planed depending on our project participations. In particular, in the context of the European TTC 2010–2012 project, we will develop UIMA wrappers for term extraction and alignement tools as well as UIMA collection management tools.
These resources and services have been set up by the Computer Sciences Laboratory of Nantes Atlantic (LINA), we invite anyone who wants to contribute to come and discuss in the `uima-fr.org` discussion list. Initially, due to some projects, the community was meant to be in Natural Language Processing and Speech Recognizing domains but it is widely open to any unstructured data management with UIMA issues.

Acknowledgements

Currently, the research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement no 248005.

9. References

David Ferrucci and Adam Lally. 2004. Uima: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348.

Iryna Gurevych, Max Mühlhäuser, Christof Müller, Jürgen Steimle, Markus Weimer, and Torsten Zesch and. 2007. Darmstadt knowledge processing repository based on uima. In *Proceedings of the First Workshop on Unstructured Information Management Architecture at Biannual Conference of the Society for Computational Linguistics and Language Technology*, Tübingen, Germany.

Udo Hahn, Ekaterina Buyko, Katrin Tomanek, Scott Piao, John McNaught, Yoshimasa Tsuruoka, and Sophia Ananiadou. 2007. An annotation type system for a data-driven nlp pipeline. In *The LAW at ACL 2007 – Proceedings of the Linguistic Annotation Workshop*, pages 33–40. Prague, Czech Republic, June 28-29, 2007. Stroudsburg, PA: Association for Computational Linguistics.

Yoshinobu Kano, Luke McCrohon, Sophia Ananiadou, and Jun'ichi Tsujii. 2009. Integrated NLP evaluation system for pluggable evaluation metrics with extensive interoperable toolkit. In *Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing (SETQA-NLP 2009)*, pages 22–30, Boulder, Colorado, June. Association for Computational Linguistics.

[27] `code.google.com/p/advanced-uima-web-rest-server`
[28] `maven.apache.org`
[29] `2009.rmll.info`
[30] `www.lina.univ-nantes.fr/-TALN-.html`
[31] `www.apache.org/licenses/LICENSE-2.0.html`
[32] GNU General Public License `www.gnu.org/licenses/gpl.html`
[33] Creative Commons Attribution-Noncommercial-Share Alike 2.0 France License `creativecommons.org/licenses/by-nc-sa/2.0/fr`
[34] GNU Free Documentation License, `www.gnu.org/licenses/fdl-1.2.html`

Peter Kluegl, Martin Atzmueller, and Frank Puppe. 2009. Textmarker: A tool for rule-based information extraction. In Christian Chiarcos, Richard Eckart de Castilho, and Manfred Stede, editors, *Proceedings of the Biennial GSCL Conference 2009, 2nd UIMA@GSCL Workshop*, pages 233–240. Gunter Narr Verlag.

Karin Verspoor, William Baumgartner Jr., Christophe Roeder, and Lawrence Hunter. 2009. Abstracting the types away from a uima type system. In *2nd UIMA Workshop at Gesellschaft für Sprachtechnologie und Computerlinguistik (GSCL)*, Tagung, Germany, October.