The MultiTal NLP tool infrastructure
Driss Sadoun, Satenik Mkhitaryan, Damien Nouvel, Mathieu Valette

To cite this version:

HAL Id: hal-01425728
https://hal-inalco.archives-ouvertes.fr/hal-01425728
Submitted on 9 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
The MultiTal NLP tool infrastructure

Driss Sadoun, Satenik Mkhitaryan, Damien Nouvel, Mathieu Valette
ERTIM, INALCO, Paris, France
firstname.lastname@inalco.fr

Abstract

This paper gives an overview of the MultiTal project, which aims to create a research infrastructure that ensures long-term distribution of NLP tools descriptions. The goal is to make NLP tools more accessible and usable to end-users of different disciplines. The infrastructure is built on a meta-data scheme modelling and standardising multilingual NLP tools documentation. The model is conceptualised using an OWL ontology. The formal representation of the ontology allows us to automatically generate organised and structured documentation in different languages for each represented tool.

1 Introduction

The work reported in this paper is initiated by INALCO (National Institute for Oriental Languages and Civilisations, also called Langues'O), a Paris-based institution for teaching and researching. It spans about 95 languages from Central Europe, Africa, Asia, America and Oceania. Historically devoted to the teaching of languages and cultures, INALCO also conducts theoretical and applied research on languages. In the context of globalisation, some new language communities indeed require to access the information-based society and the Internet. Due to the lack of responsive services, these communities have difficulties accessing language resources for their respective languages and are forced to use one of the lingua franca already well established on the Internet. As a side-effect, this also raises the risk of language impoverishment. Yet, making those languages exist on the Internet is now a necessary step for the sustainability of language diversity. Language localisation on the Internet is not only an economic issue but also a social and cultural one. Faced with the rapid growth of demand for NLP technologies, we have started the project MultiTAL (http://multital.inalco.fr:2230) of systemic description of tools processing different languages in order (i) to promote and ease the accessibility of NLP tools, (ii) to document them, (iii) finally, to plan technology transfer from one language to another.

The stakes of such a challenge are many. First, humanities and social sciences have to deal with a deep change given the increasing disaffection of students and young researchers for their disciplines. The digitisation of patrimonial funds and the emergence of new forms of communication, culture and entertainment (gaming, social networks, etc.) help in opening up new research issues. Digital humanities (DH) is the credible response to those changes. INALCO, as one of the main stakeholders in language and culture studies in Europe, faces the gap between, on one hand, the fast evolution of new technology for a few number of cultures, and, the other hand, the richness and diversity of cultures left behind the technological progress. Moreover, economic demand for localisation of products leads us to offer linguistic solutions to solve it (eg. automatic translation).

Thus, complementing the already rich offerings by looking at existing NLP tools, our aim is to offer an easy-access expert service to an accurate and critical documentation for a selected set of NLP tools and languages in our scope, rather than providing a long list of tools for well-resourced languages -but not always verified, except by the author.

1This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/
What we mean by critical documentation is building a general framework and a standard for documentation, and testing its implementation on various NLP tools. In practice, the framework has been designed using an ontology. By formalising and standardising the documentation, we aim first at designing a kind of best practice guideline for tool developers; second, at making it easier for such documentation to be read and set up for new potential beneficiaries i.e. non-expert users, for instance, linguists, students and scholars in DH, but also opinion analysts, companies that aim to enter new foreign markets, etc. - in brief, all the people who have to process foreign languages with weakly documented NLP tools. Thus, we require descriptions to be as concise, understandable and uniform as possible.

We focused first on a set of the so-called Eastern languages: here, Arabic, Chinese, Hindi, Hungarian, Japanese, Russian, Tibetan, etc. These languages present major interests for our purpose: they use different writing systems (logographic, alphabetic, etc.) which are also typographic challenges; they come from various linguistic families (Indo-European, Japonic, Semitic, Sino-Tibetan) and, even if for some they are not, strictly speaking, under-resourced languages, the tools for processing them are not always easy to handle.

Another benefit is that properly storing structured information related to NLP tools will greatly facilitate automatic generation of their descriptions. As such, our project is constrained on both aspects: relevant existing information about tools has to be saved into our inventory in order to generate concise and informative documentation.

2 Related work

Over the last few years, the number of digitized materials has considerably grown. The willingness to take into account this new digital content has led to the popularization of the use of Language Resources (LR) and NLP technologies. However, LR are still difficult to find because they are drowned in the mass of web content. Moreover, their documentation is often monolingual and written either in the developers’ language (such as Arabic, Chinese, Japanese or Russian) or in a lingua franca (such as English or French) (cf. section 3.1). This situation makes it difficult for scholars to use or re-use LR that could be useful for their work or research. Hence, storing and distributing LR has become an issue in itself. This has been addressed by many initiatives all around the world as the CLARIN projet (Váradi et al., 2008), the Central and South-East European Resources (CESAR) (VÁRADI and TADIĆ, 2012) which is a part of META-SHARE (Piperidis, 2012), the INESS Norwegian infrastructure for the construction and exploration of treebanks (Rosén et al., 2012), the large scale database SHACHI for collecting LR in Asian and Western countries (Tohyama et al., 2008), the Digital Research Infrastructure for the Arts and Humanities (Tonne et al., 2013) or the LRE MAP (Calzolari et al., 2012). These initiatives are essential to promote the research and development of language technologies. They also may provide a real picture of tools and resources that are currently available for several languages (Skadina et al., 2013; TADIĆ, 2012; Del Gratta et al., 2014). Collecting and documenting LR makes them more accessible. However, regarding NLP tools it does not necessarily makes them more usable. Our approach focuses on detailing NLP tools usage from their installation to their execution.

In order to describe and share LR, different meta-data models have been proposed (Gavrilidou et al., 2011; Broeder et al., 2012; McCrae et al., 2015a). The models of each provider depend on their coverage and the kind of LRs they manage. Hence, there are as many meta-data models for describing LRs as LR infrastructures, which may represent a limit for resource sharing and lead to the re-creation of already existing LR resources (Cieri et al., 2010). To address this issue, different attempts have been made, such as an initiative for harmonising between ELRA and LDC catalogs (Cieri et al., 2010) and more recently ontologies were used to devise interconnections among resources (Chiarosc, 2012) or to make meta-data available from different sources under a common scheme (McCrae et al., 2015a; McCrae et al., 2015b). In the perspective of an interoperability between our meta-data model and the existing ones, and in order to ease a possible integration into large infrastructure as CLARIN or META-NET we chose to use an ontology for storing MultiTal infrastructure data. The resulting triple store is accessible and freely available at http://multital.inalco.fr:2230.

Most existing LR infrastructures focus on EU languages and invite developers of resources or tools
to describe them themselves. Even if it eases access to LR technologies, when it concerns NLP tools it
does not necessarily make their use any easier. Indeed, most of the time their usage instructions remain
too poorly documented. In our project, we ambition to inventory NLP tools processing written non-
EU languages or more precisely languages taught at the INALCO. In this framework, each NLP tool is
identified, tested and fully documented by an intern speaking the language the tool processes. Then,
if the tool appears to run correctly its information is stored within our meta data model (ontology) and
its resulting documentation is made available. Our aim is to ensure that tools described on MultiTal
infrastructure can be properly installed and executed by end-users. As MultiTal’s end-users may not
be language technology experts and their mother tongue may vary, we use an ontology verbalisation
method (Androutsopoulos et al., 2014; Cojocaru and Trăuian Matu, 2015; Keet and Khumalo, 2016)
detailed in (Sadoun et al., 2016) to automatically produce documentation in multiple languages. So that
we provide end-users with simple, structured and organised documents containing NLP tool information
and detailing instructions of how to install, configure and run tools fitting their needs.

3 MultiTal infrastructure

Nowadays, language technologies (LT) make it possible for scholars to analyze millions of documents in
multiple languages with very limited manual intervention. However, retrieving and using appropriate LT
is not always easy. The MultiTal infrastructure is designed to help scholars to integrate NLP technology
into their activities. This by easing their access to, and their understanding of, NLP tools’ usage. Tools
described within the infrastructure are those that have been previously tested (cf. section 3.1). The reason
is twofold: first, to promote tools that run satisfactorily, given that some of those found on the net are
prototypes that may be obsolete or unfinished. Second, a major part of tools are designed by researchers
or individuals who are not expert at tool packaging. Hence, even tools that run correctly may be poorly
documented and so be difficult to install and execute even for an expert. Testing them allows us to dive
into the difficulties that may arise and then formalize within our model (cf. section 3.2) the different steps
of installation and execution procedures. Once a tool description is formalized within the ontology, we
automatically generate a concise and structured description of the tool containing, among other things,
the basic instructions that the user should execute to install and use it (cf. section 3.3).

Figure 1 gives an overall picture of the general data flow of MultiTal project. First, NLP interns
speaking different languages capture information about NLP tools from existing web documentation
and from what they learned by testing them. The gathered information is filled via a web platform
and stored within an ontology. Ontology knowledge is then easily retrievable through a platform that
provides fully documented NLP tools’ descriptions. Moreover, the conceptualized information serves
the automatic generation of multilingual documentations which are freely available for scholars via the
MultiTal platform.

![General data flow of MultiTal project](image)

Figure 1: General data flow of MultiTAL project

3.1 Tool documentation protocol

Over the last years, the number of NLP tools has considerably grown. However, as NLP tools are often
developed by lone researchers, in the framework of temporary projects (PhD theses, funded projects)
they do not always follow guidelines and good practices for documenting their tools. Indeed, NLP tool developers are not expert at packaging and promoting their own tools. Thus, most of the produced tools are under-described (often for personal use) and often entirely or mostly described in a monolingual documentation either in developers’ native language or in English. For example, among the selected NLP tools documented in our platform, 77% of them had a documentation available only in a single language (cf. Table 1). Hence, available documentation for NLP tool usage is not always comprehensive or well structured and may be quite complicated for a non-expert which makes it difficult for non native speakers of that language to use the tool. This leads to a limited use of such NLP tools. *MultiTal* project aims at overcoming these drawbacks.

*MultiTal* project is highly multilingual by nature as it is aimed at scholars working on various languages: our assumption is that NLP tools processing a (or various) language(s) should provide well formed documentation for multiple languages to facilitate and foster their use. Indeed, it often happens that scholars want to use NLP tools for processing documents written in a language they do not fluently speak.

Within our framework, NLP Tools are selected for a given language and according to the NLP tasks they perform. We consider any repositories from academic, institutional or personal websites. Before being added to our inventory, each NLP tool is tested in order to ensure that it can be installed and executed properly. Table 1 shows that more than a half of tested tools were not kept by our interns. NLP tools processing a language are systematically tested by an intern speaking that language. Testing is a very valuable step because it allows for instance the intern to check on which operating systems (OS) the tool can run and which are the requirements to make it run correctly. The intern may have the possibility to offer some simple patches if the tool has bugs that can be easily fixed, as for instance: adding encoding declaration, correcting typos in the execution command or in function names, pinpointing intermediate steps omitted in the original documentation, etc. Moreover, it can detail installation and execution procedures as atomic instructions that end-users must perform in order to run the documented tool. Then, all gathered information is entered via a web platform and stored in an ontology that formalises our meta-data model. For example, our Russian intern retrieved 13 tools processing Russian (cf. Table 1), 9 of them were documented only in Russian. For each of them, the intern has tested their installation and execution procedures and their ability to actually process Russian. Then she provided detailed and structured information which is formalised according to our meta-data model (cf. section 3.2) which enables us to automatically generate documentation from the model in multiple languages -such as English and French (cf. section 3.3).

<table>
<thead>
<tr>
<th>Language</th>
<th>selected</th>
<th>rejected</th>
<th>Monolingual documentation</th>
<th>Multilingual documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>25</td>
<td>15</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>Chinese</td>
<td>16</td>
<td>34</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>Hindi</td>
<td>14</td>
<td>4</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>Hungarian</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Japanese</td>
<td>14</td>
<td>19</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Marathi</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Russian</td>
<td>13</td>
<td>15</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>Tibetan</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>92</strong></td>
<td><strong>94</strong></td>
<td><strong>71</strong></td>
<td><strong>21</strong></td>
</tr>
</tbody>
</table>

Table 1: NLP tools documentation within the platform.

### 3.2 MultiTal meta-data model

To be effective, our meta-data model of NLP documentation should contain all the information needed by an NLP tool user. The kinds of information that should be included in tool documentation is typically those that should be include in a *ReadMe* file: a simple and short written document that is distributed along with a piece of software. It is written by the developer and is supposed to contain basic, crucial
information that the user should know before running the software. Writing a clear Readme file is essential for effective software distribution and use: a confusing one could prevent the user from using the tool. To our knowledge there are no established best practices for writing a ReadMe. So, in order to determine what kinds of information should be included, we proceeded to a joint study of:

1. NLP tool documentation for various languages (Chinese, English, French, Japanese, Tibetan, Hindi, Russian, etc.) that we have already tested.

2. Structured ReadMe files (more than fifty thousands) crawled from GitHub repositories.

3. Other meta-data models as META-SHARE (Gavrilidou et al., 2011) or CMDI (Broeder et al., 2012).

This study allowed us to identify the most frequent and pertinent information used to document an NLP tool. We based the conceptualization of the ontology representing our meta-data model on these results. As done for the META-SHARE and CMDI meta-data models, we define bundles of properties (super properties). These properties define the characteristics of an NLP tool such as its name, its date of creation, its affiliation (author, institution, project), its license, the system configuration on which it could run, its installation procedures or the tasks it performs, etc. Figure 2 details the conceptualized bundles of properties, together with examples of some sub-properties for the bundles Affiliation and Task. Currently, the ontology contains 46 concepts, 52 object properties and 167 data type properties.

![Figure 2: A piece of the ontology properties](image)

We distinguish two levels of meta-data: a mandatory level which provides everything the user should know before using the tool, i.e. basic elements that will commonly form a ReadMe; and a non mandatory level that contains descriptions which could be helpful to relate the tool to other tools, labs, methods or projects. Properties depicted in Figure 2 are all of the mandatory level.

The originality of the proposed model is that it focuses on NLP tools usage. The aim is to promote the use of language technologies within communities which are not familiar with their use. Hence, the model is task- and language-oriented as the choice of an NLP tool depends mostly on these two features. As a result, tools are characterized by the tasks they can perform on a given language. Indeed, tools may perform different NLP tasks and for the same tool processed languages may vary according to each task. Moreover, a task may have several execution procedures depending on the targeted language, the OS, the system configuration, the terminal or graphical mode and so on. In addition, a tool may have several installation procedures which depend also on the same factors. These procedures represent the core information for running a tool. As they can be long and tedious, it is important to describe them in a simple and comprehensive way. To do so, in our conceptual model, both installation and execution procedures are divided in their atomic instructions. Thus, when an intern enters a new procedure via the web platform, the procedure is split into atomic instructions. Each instruction is then conceptualised as triple <subject - property - object>. Figure 3 depicts an example of how an installation procedure (IP) of the Morphological Analyzer & Stemmer Darwish is conceptualised within the ontology. First, the procedure is split into its atomic instructions (on the left of the figure). Then each instruction is
conceptualised as a semantic triplet (on the middle of the figure). In addition, instructions of a procedure are numbered in order to be ordered when translated from the ontology to a documentation in a targeted language. As for the French translation depicted on the right part of the figure. Indeed, the final aim is to provide end-users structured multilingual documentation detailing the different installation and execution procedures that an NLP tool may be characterised by. The automatic generation of multilingual documentation is discussed in the next section.

The produced ontology is downloadable and queryable via SPARQL from the MultiTal infrastructure interface. From a medium-term perspective, we plan to provide automatically generated executable scripts for tools installation and execution for different OS. Moreover, SPARQL queries allow us to identify compatible NLP tools in terms of tasks, languages, OS, inputs, outputs, etc. such that they can be associated in parallel or in pipeline to improve or compare their performance.

wget https://sites.google.com/site/nlp4arabic/Darwish.zip
unzip Darwish.zip

cd Darwish

tar -zxvf sebawai.tar.gz

Figure 3: Conceptualisation of an installation procedure for the NLP tool Darwish.

3.3 Automatic generation of multilingual documentation

Before using any tool, it is generally recommended and sometimes unavoidable to read its documentation. To be understandable, this documentation should indeed be in a language that the user can read. When it comes to software products, this documentation is often called a ReadMe file (cf. Section 3.2).

In our framework, we focus on NLP tools processing languages that are taught at INALCO. These languages are for most of them not lingua franca. Till now, for the eight languages we investigated, 77% of the selected tools were documented only in the developer’s mother tongue or in English (cf. Table 1). Otherwise, a version in English or French was also available. So, in order to ease and improve the experience of end-users, the MultiTal infrastructure aims at providing multilingual documentation for NLP tools processing different languages. To do so, we use an ontology verbalisation approach detailed in (Sadoun et al., 2016) that benefits from the formalization of NLP tools information to automatically generate multilingual ReadMe files that contain simplified and structured information about each tool such as its license, its installation, execution or training instructions or the language it processes etc.

3.4 MultiTal infrastructure in practice

The MultiTal project is conducted at the INALCO institution which is a crossroads for Languages and Civilizations. It hosts students, lecturers and researchers of several disciplines from all around the world practicing almost one hundred different languages. Many of these scholars are confronted to a constantly increasing number of digitized data, so that the use of NLP technologies becomes more and more valuable for their practice. MultiTal infrastructure is dedicated to making such technologies more accessible regardless of the expertise or spoken language of end-users.

Currently, the infrastructure contains documentation for 92 NLP tools. These documented tools perform 202 NLP tasks of 46 different types. These documented tools perform 202 NLP tasks of 46 different types. They are distributed across more than eight languages as some of them process more than one language. Though, for each tool a distinction is made between the languages it manages that have been tested and those that have not been tested yet.

The MultiTal infrastructure website is currently available in seven languages (English, French, Spanish, Chinese, Russian, Arabic and Japanese). Figure 4 shows a fragment of the research interface. Selection of tools can be made according to the NLP task they perform, the language they manage, their developer(s), their institution, the way they are accessible (downloadable, on-line or web-service), etc. In addition, NLP tools’ documentation inventory gives us key information and statistics. For example,
the Pie chart depicted in Figure 5 details the distribution of NLP tasks performed by all the documented tools. On the chart, we can see the eight most performed tasks. We can also produce statistics regarding each language (or family) of languages to see, for example, how advanced those languages are in terms of NLP processing and whether they are under-resourced languages or not.

Finally, we count on INALCO’s scholars diversity to make the infrastructure grow. Our scholars will both have the benefit of the multilingual NLP documentation provided by the MultiTal infrastructure and the opportunity to help us to make it evolve.

4 Conclusion

For a typical scholar, finding NLP tools relevant to their need is not as easy as it should be. Even when relevant NLP tools are found it appears that they are not that simple to use. The MultiTal infrastructure is initiated to overcome this situation. In this paper, we described the MultiTal infrastructure meta-data model based on the use of an ontology. We motivated our choice which aims to ease and foster the use of NLP tools by scholars of different disciplines. In a short term perspective, we plan to evaluate the effectiveness of the produced documentation, to see whether it provides all the needed information and if it is easy to follow for non NLP experts.

In the future, by considering the produced expertise about NLP tools, we ought to be able to develop methods for adapting some of the tools to languages they have not been designed for, by training them. Indeed, alongside tool identification, we collect information about tagged corpora in order to use them as training ones. Finally, the formalisation of execution procedures into their atomic instructions already allows us to run execution scripts. We intend to use these scripts to combine the execution of different NLP tools either in pipelines or in parallel in order to compare and/or increase their performance.
References


