Language Resources Factory: case study on the acquisition of Translation Memories*

Marc Poch
UPF Barcelona, Spain
marc.pochriera@upf.edu

Antonio Toral
DCU Dublin, Ireland
atoral@computing.dcu.ie

Núria Bel
UPF Barcelona, Spain
nuria.bel@upf.edu

Abstract

This paper demonstrates a novel distributed architecture to facilitate the acquisition of Language Resources. We build a *factory* that automates the stages involved in the acquisition, production, updating and maintenance of these resources. The factory is designed as a platform where functionalities are deployed as web services, which can be combined in complex acquisition chains using workflows. We show a case study, which acquires a Translation Memory for a given pair of languages and a domain using web services for crawling, sentence alignment and conversion to TMX.

1 Introduction

A fundamental issue for many tasks in the field of Computational Linguistics and Language Technologies in general is the lack of Language Resources (LRs) to tackle them successfully, especially for some languages and domains. It is the so-called LRs bottleneck.

Our objective is to build a factory of LRs that automates the stages involved in the acquisition, production, updating and maintenance of LRs required by Machine Translation (MT), and by other applications based on Language Technologies. This automation will significantly cut down the required cost, time and human effort. These reductions are the only way to guarantee the continuous supply of LRs that Language Technologies demand in a multilingual world.

* We would like to thank the developers of Soaplab, Taverna, myExperiment and Biocatalogue for solving our questions and attending our requests. This research has been partially funded by the EU project PANACEA (7FP-ICT-248064).

2 Web Services and Workflows

The factory is designed as a platform of web services (WSs) where the users can create and use these services directly or combine them in more complex chains. These chains are called workflows and can represent different combinations of tasks, e.g. "extract the text from a PDF document and obtain the Part of Speech (PoS) tagging" or "crawl this bilingual website and align its sentence pairs". Each task is carried out using NLP tools deployed as WSs in the factory.

Web Service Providers (WSPs) are institutions (universities, companies, etc.) who are willing to offer services for some tasks. WSs are services made available from a web server to remote users or to other connected programs. WSs are built upon protocols, server and programming languages. Their massive adoption has contributed to make this technology rather interoperable and open. In fact, WSs allow computer programs distributed in different locations to interact with each other.

WSs introduce a completely new paradigm in the way we use software tools. Before, every researcher or laboratory had to install and maintain all the different tools that they needed for their work, which has a considerable cost in both human and computing resources. In addition, it makes it more difficult to carry out experiments that involve other tools because the researcher might hesitate to spend time resources on installing new tools when there are other alternatives already installed.

The paradigm changes considerably with WSs, as in this case only the WSP needs to have a deep knowledge of the installation and maintenance of the tool, thus allowing all the other users to benefit

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 1–5,
Avignon, France, April 23 - 27 2012.

from this work. Consequently, researchers think about tools from a high level and solely regarding their functionalities, thus they can focus on their work and be more productive as the time resources that would have been spent to install software are freed. The only tool that the users need to install in order to design and run experiments is a WS client or a Workflow editor.

3 Choosing the tools for the platform

During the design phase several technologies were analyzed to study their features, ease of use, installation, maintenance needs as well as the estimated learning curve required to use them. Interoperability between components and with other technologies was also taken into account since one of our goals is to reach as many providers and users as possible. After some deliberation, a set of technologies that have proved to be successful in the Bioinformatics field were adopted to build the platform. These tools are developed by the myGrid[1] team. This group aims to develop a suite of tools for researchers that work with e-Science. These tools have been used in numerous projects as well as in different research fields as diverse as astronomy, biology and social science.

3.1 Web Services: Soaplab

Soaplab (Senger et al., 2003)[2] allows a WSP to deploy a command line tool as a WS just by writing a metadata file that describes the parameters of the tool. Soaplab takes care of the typical issues regarding WSs automatically, including temporary files, protocols, the WSDL file and its parameters, etc. Moreover, it creates a Web interface (called Spinet) where WSs can be tested and used with input forms. All these features make Soaplab a suitable tool for our project. Moreover, its numerous successful stories make it a safe choise; e.g., it has been used by the European Bioinformatics Institute[3] to deploy their tools as WSs.

3.2 Registry: Biocatalogue

Once the WSs are deployed by WSPs, some means to find them becomes necessary. Biocatalogue (Belhajjame et al., 2008)[4] is a registry where WSs can be shared, searched for, annotated with tags, etc. It is used as the main registration point for WSPs to share and annotate their WSs and for users to find the tools they need. Biocatalogue is a user-friendly portal that monitors the status of the WSs deployed and offers multiple metadata fields to annotate WSs.

3.3 Workflows: Taverna

Now that users can find WSs and use them, the next step is to combine them to create complex chains. Taverna (Missier et al., 2010)[5] is an open source application that allows the user to create high-level workflows that integrate different resources (mainly WSs in our case) into a single experiment. Such experiments can be seen as simulations which can be reproduced, tuned and shared with other researchers.

An advantage of using workflows is that the researcher does not need to have background knowledge of the technical aspects involved in the experiment. The researcher creates the workflow based on functionalities (each WS provides a function) instead of dealing with technical aspects of the software that provides the functionality.

3.4 Sharing workflows: myExperiment

MyExperiment (De Roure et al., 2008)[6] is a social network used by workflow designers to share workflows. Users can create groups and share their workflows within the group or make them publically available. Workflows can be annotated with several types of information such as description, attribution, license, etc. Users can easily find examples that will help them during the design phase, being able to reuse workflows (or parts of them) and thus avoiding *reinveinting the wheel*.

4 Using the tools to work with NLP

All the aforementioned tools were installed, used and adapted to work with NLP. In addition, several tutorials and videos have been prepared[7] to help partners and other users to deploy and use WSs and to create workflows.

Soaplab has been modified (a patch has been developed and distributed)[8] to limit the amount of data being transfered inside the SOAP message in

[1] http://www.mygrid.org.uk
[2] http://soaplab.sourceforge.net/soaplab2/
[3] http://www.ebi.ac.uk
[4] http://www.biocatalogue.org/
[5] http://www.taverna.org.uk/
[6] http://www.myexperiment.org/
[7] http://panacea-lr.eu/en/tutorials/
[8] http://myexperiment.elda.org/files/5

order to optimize the network usage. Guidelines that describe how to limit the amount of concurrent users of WSs as well as to limit the maximum size of the input data have been prepared.[9]

Regarding Taverna, guidelines and workflow examples have been shared among partners showing the best way to create workflows for the project. The examples show how to benefit from useful features provided by this tool, such as "retries" (to execute up to a certain number of times a WS when it fails) and "parallelisation" (to run WSs in parallel, thus increasing trhoughput). Users can view intermediate results and parameters using the provenance capture option, a useful feature while designing a workflow. In case of any WS error in one of the inputs, Taverna will report the error message produced by the WS or processor component that causes it. However, Taverna will be able to continue processing the rest of the input data if the workflow is robust (i.e. makes use of retry and parallelisation) and the error is confined to a WS (i.e. it does not affect the rest of the workflow).

An instance of Biocatalogue and one of myExperiment have been deployed to be the Registry and the portal to share workflows and other experiment-related data. Both have been adapted by modifying relevant aspects of the interface (layout, colours, names, logos, etc.). The categories that make up the classification system used in the Registry have been adapted to the NLP field. At the time of writing there are more than 100 WSs and 30 workflows registered.

5 Interoperability

Interoperability plays a crucial role in a platform of distributed WSs. Soaplab deploys SOAP[10] WSs and handles automatically most of the issues involved in this process, while Taverna can combine SOAP and REST[11] WSs. Hence, we can say that communication protocols are being handled by the tools. However, parameters and data interoperability need to be addressed.

5.1 Common Interface

To facilitate interoperability between WSs and to easily exchange WSs, a Common Interface (CI) has been designed for each type of tool (e.g. PoS-taggers, aligners, etc.). The CI establishes that all WSs that perform a given task must have the same mandatory parameters. That said, each tool can have different optional parameters. This system eases the design of workflows as well as the exchange of tools that perform the same task inside a workflow. The CI has been developed using an XML schema.[12]

5.2 Travelling Object

A goal of the project is to facilitate the deployment of as many tools as possible in the form of WSs. In many cases, tools performing the same task use in-house formats. We have designed a container, called "Travelling Object" (TO), as the data object that is being transfered between WSs. Any tool that is deployed needs to be adapted to the TO, this way we can interconnect the different tools in the platform regardless of their original input/output formats.

We have adopted for TO the XML Corpus Encoding Standard (XCES) format (Ide et al., 2000) because it was the already existing format that required the minimum transduction effort from the in-house formats. The XCES format has been used successfully to build workflows for PoS tagging and alignment.

Some WSs, e.g. dependency parsers, require a more complex representation that cannot be handled by the TO. Therefore, a more expressive format has been adopted for these. The Graph Annotation Format (GrAF) (Ide and Suderman, 2007) is a XML representation of a graph that allows different levels of annotation using a "feature–value" paradigm. This system allows different in-house formats to be easily encapsulated in this container-based format. On the other hand, GrAF can be used as a pivot format between other formats (Ide and Bunt, 2010), e.g. there is software to convert GrAF to UIMA and GATE formats (Ide and Suderman, 2009) and it can be used to merge data represented in a graph.

Both TO and GrAF address syntactic interoperability while semantic interoperability is still an open topic.

[9] http://myexperiment.elda.org/files/4

[10] http://www.w3.org/TR/soap/

[11] http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

[12] http://panacea-lr.eu/en/info-for-professionals/documents/

6 Evaluation

The evaluation of the factory is based on its features and usability requirements. A binary scheme (yes/no) is used to check whether each requirement is fulfilled or not. The quality of the tools is not altered as they are deployed as WSs without any modification. According to the evaluation of the current version of the platform, most requirements are fulfilled (Aleksić et al., 2012).

Another aspect of the factory that is being evaluated is its performance and scalabilty. They do not depend on the factory itself but on the design of the workflows and WSs. WSPs with robust WSs and powerful servers will provide a better and faster service to users (considering that the service is based on the same tool). This is analogous to the user installing tools on a computer; if the user develops a fragile script to chain the tools the execution may fail, while if the computer does not provide the required computational resources the performance will be poor.

Following the example of the Bioinformatics field where users can benefit of powerful WSPs, the factory is used as a proof of concept that these technologies can grow and scale to benefit many users.

7 Case study

We introduce a case study in order to demonstrate the capabilities of the platform. It regards the acquisition of a Translation Memory (TM) for a language pair and a specific domain. This is deemed to be very useful for translators when they start translating documents for a new domain. As at that early stage they still do not have any content in their TM, having the automatically acquired TM can be helpful in order to get familiar with the characteristic bilingual terminology and other aspects of the domain. Another obvious potential use of this data would be to use it to train a Statistical MT system.

Three functionalities are needed to carry out this process: acquisition of the data, its alignment and its conversion into the desired format. These are provided by WSs available in the registry.

First, we use a domain-focused bilingual crawler[13] in order to acquire the data. Given a pair of languages, a set of web domains and a set of seed terms that define the target domain for these languages, this tool will crawl the webpages in the domains and gather pairs of web documents in the target languages that belong to the target domain. Second, we apply a sentence aligner.[14] It takes as input the pairs of documents obtained by the crawler and outputs pairs of equivalent sentences.Finally, convert the aligned data into a TM format. We have picked TMX[15] as it is the most common format for TMs. The export is done by a service that receives as input sentence-aligned text and converts it to TMX.[16]

The "Bilingual Process, Sentence Alignment of bilingual crawled data with Hunalign and export into TMX"[17] is a workflow built using Taverna that combines the three WSs in order to provide the functionality needed. The crawling part is ommitted because data only needs to be crawled once; crawled data can be processed with different workflows but it would be very inefficient to crawl the same data each time. A set of screenshots showing the WSs and the workflow, together with sample input and output data is available.[18]

8 Demo and Requirements

The demo aims to show the web portals and tools used during the development of the case study. First, the Registry[19] to find WSs, the Spinet Web client to easily test them and Taverna to finally build a workflow combining the different WSs. For the live demo, the workflows will be already designed because of the time constraints. However, there are videos on the web that illustrate the whole process. It will be also interesting to show the myExperiment portal,[20] where all public workflows can be found. Videos of workflow executions will also be available.

Regarding the requirements, a decent internet connection is critical for an acceptable performance of the whole platform, specially for remote WSs and workflows. We will use a laptop with Taverna installed to run the workflow presented in Section 7.

[13] http://registry.elda.org/services/127

[14] http://registry.elda.org/services/92

[15] http://www.gala-global.org/oscarStandards/tmx/tmx14b.html

[16] http://registry.elda.org/services/219

[17] http://myexperiment.elda.org/workflows/37

[18] http://www.computing.dcu.ie/~atoral/panacea/eacl12_demo/

[19] http://registry.elda.org

[20] http://myexperiment.elda.org

References

Vera Aleksić, Olivier Hamon, Vassilis Papavassiliou, Pavel Pecina, Marc Poch, Prokopis Prokopidis, Valeria Quochi, Christoph Schwarz, and Gregor Thurmair. 2012. Second evaluation report. Evaluation of PANACEA v2 and produced resources (PANACEA project Deliverable 7.3). Technical report.

Khalid Belhajjame, Carole Goble, Franck Tanoh, Jiten Bhagat, Katherine Wolstencroft, Robert Stevens, Eric Nzuobontane, Hamish McWilliam, Thomas Laurent, and Rodrigo Lopez. 2008. Biocatalogue: A curated web service registry for the life science community. In *Microsoft eScience conference*.

David De Roure, Carole Goble, and Robert Stevens. 2008. The design and realisation of the myexperiment virtual research environment for social sharing of workflows. *Future Generation Computer Systems*, 25:561–567, May.

Nancy Ide and Harry Bunt. 2010. Anatomy of annotation schemes: mapping to graf. In *Proceedings of the Fourth Linguistic Annotation Workshop*, LAW IV '10, pages 247–255, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nancy Ide and Keith Suderman. 2007. GrAF: A Graph-based Format for Linguistic Annotations. In *Proceedings of the Linguistic Annotation Workshop*, pages 1–8, Prague, Czech Republic, June. Association for Computational Linguistics.

Nancy Ide and Keith Suderman. 2009. Bridging the Gaps: Interoperability for GrAF, GATE, and UIMA. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 27–34, Suntec, Singapore, August. Association for Computational Linguistics.

Nancy Ide, Patrice Bonhomme, and Laurent Romary. 2000. XCES: An XML-based encoding standard for linguistic corpora. In *Proceedings of the Second International Language Resources and Evaluation Conference. Paris: European Language Resources Association*.

Paolo Missier, Stian Soiland-Reyes, Stuart Owen, Wei Tan, Aleksandra Nenadic, Ian Dunlop, Alan Williams, Thomas Oinn, and Carole Goble. 2010. Taverna, reloaded. In M. Gertz, T. Hey, and B. Ludaescher, editors, *SSDBM 2010*, Heidelberg, Germany, June.

Martin Senger, Peter Rice, and Thomas Oinn. 2003. Soaplab - a unified sesame door to analysis tools. In *All Hands Meeting*, September.