# HPC-ready Language Analysis for Human Beings

*Emanuele Lapponi*[1], *Erik Velldal*[1], *Nikolay A. Vazov*[2], *Stephan Oepen*[1]

(1) Language Technology Group, Department of Informatics, University of Oslo
(2) Research Support Services Group, University Center for Information Technology, University of Oslo

{emanuel|erikve|oe}@ifi.uio.no, n.a.vazov@usit.uio.no

ABSTRACT

This demonstration presents a first operable pilot of the Language Analysis Portal (LAP), an ongoing project within the Norwegian CLARINO initiative that aims at providing easy access to Language Technology (LT) tools running on a powerful High-Performance Computing (HPC) cluster. The system is built on top of the Galaxy framework, giving users an on-line platform where they can design experiments using an array of processors. These processors can be combined into complex workflows using a visual editor. The current implementation functions as a testbed for further development, hosting a limited collection of tools addressing common use-cases in the LT-realm; the long-term goal for LAP is to reach beyond the field and be an enabling platform for LT-powered research in the humanities and social sciences.

KEYWORDS: research infrastructure, High-Performance Computing, web portal, CLARINO.

# 1 Introduction

This demonstration will be showcasing ongoing work within the CLARINO[1] project on building a web portal for natural language analysis. The effort is carried out at the University of Oslo (UiO) jointly by the Language Technology Group (LTG) and the Research Computing group at the University Center for Information Technology (USIT).

An important part of the overall mission of CLARINO is to facilitate the use of language technology (LT) in the social sciences and humanities. In the same vein, the construction of the Language Analysis Portal (LAP) aims to boost the availability and usability of large-scale language analysis for researchers both within and outside the field of LT itself. Currently, many common LT tools can appear rather daunting to use, requiring a lot of technical knowledge on the side of the user. Apart from the challenge of orienting oneself in the fragmented ecosystem of available tools, many potential users, especially from less technically oriented disciplines, might not be comfortable with command-line interfaces or having to wrestle with difficult and poorly documented installation procedures, or might lack the required knowledge about annotation formats or other dependencies. Many researchers might also not have access to the computing power necessary to process larger data sets. LAP aims to eliminate such obstacles. The goal is to maintain a large repository of LT tools that are easily accessible through a web portal, offering a uniform graphical interface and ensuring a low bar of entry for users, while at the same time enabling execution of complex workflows to be run on a high-performance computing (HPC) cluster, also ensuring scalability to very large data sets. LAP's HPC-centric design also sets it apart from other abstractly similar infrastructure projects such as WebLicht[2] within CLARIN-D, in that all the tools will be hosted locally and adapted to work with the national grid infrastructure.

While the LAP development is still in its early stages, an operable pilot is already available for testing and demonstration purposes. The remaining part of the paper is structured analogously to the interactive demonstration, viz. providing a walk-through of a representative use case—touching on everything from user authentication and data set management to workflow design and viewing results—commenting on technical details along the way.

# 2 LAP Pilot

The long-term goal for LAP is to reach beyond the field and be an enabling platform for LT-powered research in the humanities and social sciences. At this early stage of development, however, the pilot use case is centered around the needs of LT researchers and in particular students, as their level of proficiency strikes a good balance between non programming-savvy historians and seasoned programmers. This allows us to make certain assumptions that facilitate the initial development stage—we assume the users have a certain knowledge of what the installed tools do—while at the same time providing test candidates that can truly benefit from the system.

The pilot system serves several purposes: Most importantly it will act as a *proof-of-concept*, assessing the viability of involved software as well as ideas before extending the implementation to a larger set of tools and support for a larger set of features. The most important part of this, in turn, is assessing the suitability of the *Galaxy platform* (Giardine et al., 2005;

[1]The Norwegian branch of the pan-European CLARIN project (Common Language Resources and Technology Infrastructure). For more information see; `http://clarin.b.uib.no/`

[2]The WebLicht website: `http://weblicht.sfs.uni-tuebingen.de/weblichtwiki/`

Blankenberg et al., 2010; Goecks et al., 2010), a web-based workflow management system initially developed for data-intensive research in genomics and bioinformatics , which is a core component of the current pilot implementation.[3] Another important use of the pilot will be as a *demonstrator*; for reaching out to tool developers, to illustrate use cases for potential user groups in the humanities and social sciences, and as a foundation for further surveying user-requirements. The pilot is only meant to support a minimal selection of LT tools and will be evaluated in part by a group of test users consisting of master students of the study program *Informatics: Language and Communication* at the University of Oslo.

In this context, a typical use case is that of annotating natural language text with syntactic analyses, e.g. annotating a collection of newspaper articles or a novel with dependency graphs. The completion of this task would ordinarily require the user to install the relevant software packages and execute the tools that satisfy the dependencies of the parser; minimally a sentence-segmenter, a tokenizer and a POS-tagger. Furthermore, the user in question could be interested in producing dependency annotations generated using different parsers, that are in turn invoked with part of speech tags that originate from different upstream annotators. These actions typically require at least basic Unix-shell proficiency; wrapping the execution in a small, parameterizable program that allows the user to effectively try out different settings further raises the entry-bar for this kind of experimental process. Additionally, different tools in the chain might use different representations—tab-separated columns or nested XML, say—requiring the user to further pre-process the data at each step. Lack of in-depth technical knowledge, perhaps in combination with lack of good documentation, might make each of these steps difficult for many potential users, especially for those from less technically oriented disciplines. Access to the necessary computing power might itself also be a barrier, depending on the complexity of the task or the size of the dataset in question.

The LAP pilot includes enough tools and functionality to enable this kind of computation with a few mouse-clicks, allowing users to visually design complex tool-chains, store results and experiment with different tool-configurations and datasets on any platform that can run a modern web-browser. After logging into the system via Feide, users are presented with the Galaxy *workspace*, as shown in Figure 1. The left panel displays the installed processing tools; clicking on tool-names displays general information and configuration options in the center panel. In order to be integrated in the LAP tool-chain, existing tools are 'wrapped' inside scripts that decode the LAP-internal format, present the tool itself with its expected input and finally re-encode the output so that it is compatible with the next processing step. For LAP's system-internal representation, we are at the moment looking into both TCF (Text Corpus Format, Heid et al., 2010) (the format used within WebLicht, which comes with a full, albeit closed-source API) and our own in-house JSON-based LTON format (Language Technology Object Notation) which is still currently under development.

Additionally, the wrapper handles the submission of the job to the Abel cluster, a shared resource for research computing boasting more than 600 machines, totaling more than 10.000 cores (CPUs).[4] The HPC connection is a very important feature, as Language technology can be computationally quite expensive, often involving sub-problems where known best solutions

[3] For more information about Galaxy, please see `http://wiki.galaxyproject.org/`

[4] At the time of writing, the cluster ranks at position 134 on the `http://www.top500.org` list of top supercomputers world- wide. Among its frequent users, besides the language technology group, we find researchers from the life sciences, astrophysics, geophysics, and chemistry.

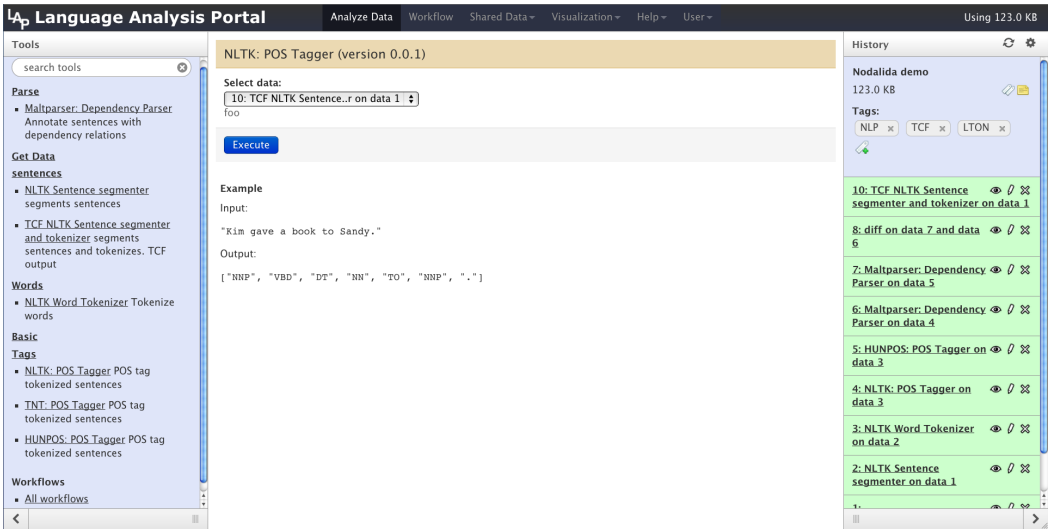


Figure 1: A screenshot of the current implementation of LAP within Galaxy. The left panel displays the installed tools, the center panel shows the currently selected function and the right panel hosts the file history.

have exponential worst-case complexity. At the same time, typical language analysis tasks can be trivially parallelized, as processing separate documents (and for many tasks also individual sentences) constitute independent units of computation. The fact that the portal will submit the sub-tasks of a workflow to an underlying HPC cluster—without the need for user knowledge about job scheduling etc.—means that the user will be able to perform analyses that might otherwise not be possible (and faster and on larger data sets).

Returning to Figure 1, the right panel contains so-called *histories*, where files associated with different experiments are collected and persistently stored across sessions. Users can add datasets to a history by uploading local files, providing a URL to the dataset location, or by pasting the relevant text using the 'Get Data' tool. The simplest action achievable with this pilot consists of adding a dataset to a history and running one of the processing tools. The processing job then starts and the annotated dataset is added to the same history; users can then select it and run other annotators. The resulting datasets can either be inspected within LAP or downloaded (the system also allows for full histories to be downloaded as compressed archives).

Most importantly, users can also compose workflows using the workflow editor, where experimental tool-chains such as the one sketched previously can be designed using a simple graphical interface; Figure 2 shows an example of workflow design within the editor, here with a workflow with four endpoints. To build the chain, users invoke GUI elements representing tool instances to the center panel and link inputs to outputs, guiding the data flow across tools. Workflows can then be saved and shared with all or selected LAP users. When running a workflow, the system populates the history with a version of the dataset for each processing point, so that the user can return to the partially processed data and run (or re-run) relevant tools without re-starting the whole workflow.

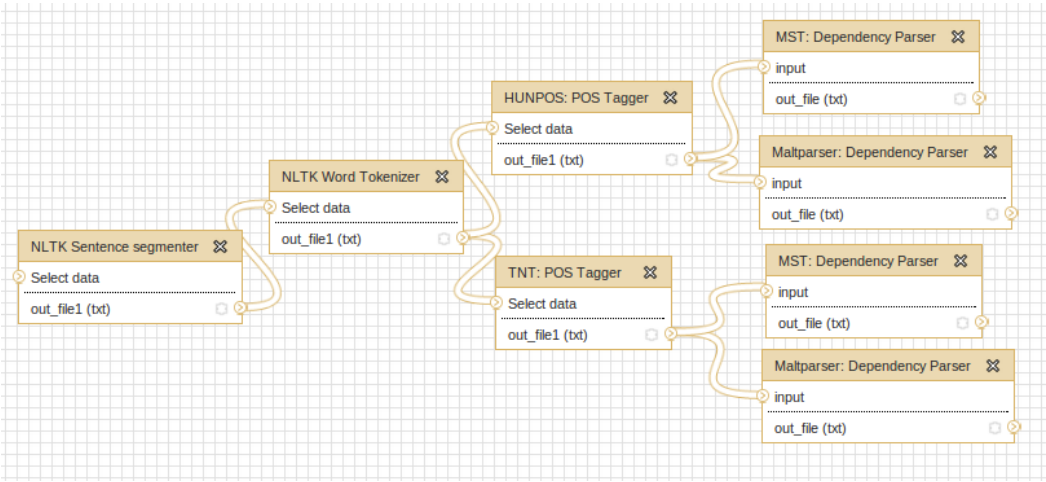


Figure 2: A user-defined LAP workflow with four endpoints. The data processing starts with unannotated text and ends with four datasets containing different annotations.

## 3 Conclusion and Outlook

In this demonstration we presented the current pilot implementation of the Language Analysis Portal (LAP), an in-development web-portal for natural language processing. LAP, developed under the umbrella of the CLARINO initiative, aims to facilitate the use of LT-tools for non-programmers by letting users create complex experimental setups visually, and allowing them to run large-scale processes on the national grid infrastructure for HPC computing. Further discussion of the technical details of the pilot can be found in (Lapponi et al., 2013).

While the long-term goal for the portal is to also address the language processing needs of researchers outside the field, the current implementation focuses on a typical use case within the LT realm, allowing users to quickly and simply combine processing tools into complex workflows and collect the results. Future releases of the systems will feature a larger array of tools, and further work starting after the release of the final pilot will address the challenge of shaping LAP into a useful research tool for the humanities and the social sciences, investigating possible use cases and surveying user-requirements from active researchers from these fields.

## References

Blankenberg, D., Kuster, G. V., Coraor, N., Ananda, G., Lazarus, R., Mangan, M., Nekrutenko, A., and Taylor, J. (2010). Galaxy: a web-based genome analysis tool for experimentalists. *Current Protocols in Molecular Biology*, pages 19.10.1–19.10.21.

Giardine, B., Riemer, C., Hardison, R. C., Burhans, R., Elnitski, L., Shah, P., Zhang, Y., Blankenberg, D., Albert, I., Taylor, J., Miller, W., Kent, W. J., and Nekrutenko, A. (2005). Galaxy: a platform for interactive large-scale genome analysis. *Genome Research*, 15(10):1451–5.

Goecks, J., Nekrutenko, A., Taylor, J., and Team, T. G. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology*, 11(8):R86.

Heid, U., Schmid, H., Eckart, K., and Hinrichs, E. (2010). A corpus representation format for linguistic web services: The D-SPIN Text Corpus Format and its relationship with ISO standards. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 494–499, Malta.

Lapponi, E., Velldal, E., Vazov, N. A., and Oepen, S. (2013). Towards large-scale language analysis in the cloud. In *Proceedings of the Workshop on Nordic Language Research Infrastructure at the 19th Nordic Conference of Computational Linguistics*, Oslo, Norway.