Towards Large-Scale Language Analysis in the Cloud

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 paper documents ongoing work within the Norwegian CLARINO project on building a Language Analysis Portal (LAP). The portal will provide an intuitive and easily accessible web interface to a centralized repository of a wide range of language technology tools, all installed on a high-performance computing cluster. Users will be able to compose and run workflows using an easy-to-use graphical interface, with multiple tools and resources chained together in potentially complex pipelines. Although the project aims to reach out to a diverse set of user groups, it particularly will facilitate use of language analysis in the social sciences, humanities, and other fields without strong computational traditions. While the development of the portal is still in its early stages, this paper documents ongoing work towards an already operable pilot in addition to providing an overview of long-term goals and visions. At the core of the current pilot implementation we find Galaxy, a web-based workflow management system initially developed for data-intensive research in genomics and bioinformatics; therefore, an important part of the work on the pilot is to adapt and evaluate Galaxy for the context of a language analysis portal.

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

1 Introduction

This paper describes ongoing work on building a web portal for natural language analysis, carried out at the University of Oslo (UiO) as a joint effort by the Language Technology Group (LTG) and the Research Computing group at the University Center for Information Technology (USIT). The work forms part of the CLARINO infrastructure initiative,[1] the Norwegian branch of the pan-European CLARIN[2] federation (Common Language Resources and Technology Infrastructure). The aim is to provide an easily accessible web interface that ensures a low bar of entry for users, while at the same time enabling execution of complex workflows and scalability to very large data sets, integrating a wide range of tools to be run on a high-performance computing (HPC) cluster. While the development of the Language Analysis Portal (LAP) is still in its early stages, the current paper documents ongoing work towards an already operable pilot, in addition to providing an overview of long-term goals and visions. A core component of the current pilot implementation is Galaxy (Giardine et al., 2005; Blankenberg et al., 2010; Goecks et al., 2010); a web-based workflow management system initially developed for data-intensive research in genomics and bioinformatics. We here document our efforts on adapting and evaluating Galaxy for the purposes of LAP.

The paper is structured as follows. Section 2 presents a high-level overview of the many aspects related to the overall vision for LAP; ranging from target user groups and interface design to technical specifications and architecture issues. Section 3 surveys other related infrastructure projects, as well as relevant processing frameworks more generally. In Section 4, we present the details of the current LAP pilot and its implementation.

2 LAP: Language Analysis Portal

The efforts described in this paper implement a workpackage of CLARINO, the Norwegian branch of the European CLARIN initiative. CLARINO is dedicated to establishing a shared research infrastructure for language technology (LT) that ensures easy access to persistent and interoperable resources and services. A particularly important part of the mission is to facilitate the use of this infrastructure in the social sciences and humanities. LAP shares this goal in that it aims to boost the availability and usability of large-scale language analysis for researchers both within and outside of the field. In this section we present a high-level view of the kinds of functionality and services that we ultimately aim for in LAP.

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. Any scholar registered in the system for federated identity management in the Norwegian education sector, Feide,[3] or the CLARIN AAI

[1] The CLARINO website: `http://clarin.b.uib.no/`

[2] The CLARIN website: `http://www.clarin.eu/`

[3] For more information, please see `https://www.feide.no/om-feide`

(Authentication and Authorization Infrastructure) will be able to log into the portal and create a user. Each user will have her own personal *workspace*, allowing data to be stored persistently across sessions. In addition to upload and storage facilities for user-provided data, the portal will also give access to common, pre-existing language resources. It will include tools for content extraction and layout analysis (from common file formats and markup schemes), as well as a comprehensive repository of language analysis tools. The portal will reach out to developers of processing tools, seeking to install the broadest possible range of technologies—ranging from token- to discourse-level analysis and encompassing both rule-based and statistical approaches. In terms of linguistic coverage, LAP will focus on languages actively used in Norway, e.g., Norwegian *Bokmål* and *Nynorsk*, Sámi, other Scandinavian languages, and English—initially at least with a focus on written language.

A central part of the interface will be a *workflow manager*, enabling the user to specify and execute a series of computations. For example, starting with a pdf-document uploaded by the user, she might further want to perform content extraction, sentence segmentation, tokenization, POS tagging, parsing, and finally identification of subjective expressions with positive polarity—all carried out in a consecutive sequence. The output of each component provides the input to the next connected component(s) in the workflow, creating a potentially complex pipeline. Note that the platform we are building on for our pilot implementation, Galaxy, comes equipped with a sophisticated workflow manager, as further described in Section 4.1. Then, after the desired workflow has been specified; at the click of a few buttons, the resources and tools involved will be configured and submitted to the national grid infrastructure, where computational and storage resources are readily available on a scale traditionally inaccessible to academic users.

This latter point, the fact that the portal will be built on top of an HPC cluster, is a crucial feature. Language technology can be computationally quite expensive, often involving sub-problems where known best solutions 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). More details about the cluster itself are presented in Section 4.3 below.

Another important requirement of the design is that it should abstract away certain low-level details—the user must be able to design and run workflows without in-depth technical knowledge of the tools or data formats involved. To make this possible the portal interface should itself have built-in awareness of inter-dependencies among component tools, standardized interchange formats and corresponding conversion procedures, etc. For example, in a given step in composing a workflow, the interface should only present the user with options that are compatible with the output of the previous step in the tool chain. Similar context-sensitive menus are also implemented in the WebLicht portal, as further described in Section 3 below.

At the same time, it is important that the abstractions provided by the interface are flexible enough to also allow for detailed control and parametrization for the more advanced users. For a given component in the workflow, the user should be able to "look under the hood" and specify parameters or options in a manner that is closer to the level of command-line interaction. The user should also have access to a recorded *history*, tracking previous actions, making it possible to reuse—or even share—workflows. Similar functionality could be used for supplying

built-in or predefined workflows corresponding to typical analysis pipelines.

One of several sources of inspiration for the ideas sketched above is the positive outcome of providing an abstractly similar portal for *computational biology*, BioPortal,[4] also developed and maintained at the University of Oslo. Internationally there are also other related infrastructure projects specifically for LT, such as WebLicht project. In the next section we take a closer look at these and other related efforts and processing frameworks.

3 Related Efforts and Relevant Frameworks

Among the infrastructure projects targeting LT, the WebLicht[5] project—developed within the German branch of CLARIN (CLARIN-D) and its predecessor project D-SPIN—is the one most relevant and similar in spirit to LAP. WebLicht implements an online environment for annotating text and constructing pipelines of various LT tools. A graphical interface allows users to upload text, convert it to a system-internal exchange format, build processing pipelines using the available tools and finally visualize and download the results. The tools offered through WebLicht are distributed across repositories maintained by various CLARIN partners, operating as synchronous REST-style web services. The system-internal representation for exchanging annotations between various components in the tool chain is an XML-based format called TCF (Text Corpus Format, Heid et al., 2010), developed within WebLicht. LAP's HPC-centric design sets it apart from the current implementation of WebLicht, in that all the tools will be hosted locally and adapted to work with the national grid infrastructure. Additionally, given the generous storage and processing facilities available, LAP will allow users to upload and annotate large datasets.

Another CLARIN initiated effort is that of CLARIN-ES-LAB, where a collection of text processing tools have been made available as web services using Soaplab. [6] Finally, the Language Grid[7] initiative should also be mentioned, a Japanese analogue to WebLicht, with a focus on machine translation tools.

Another data intensive field of research that has seen the need for online portal services for computationally demanding applications is that of bioinformatics. Notably, one such portal, BioPortal, is developed and hosted at the University of Oslo. Offering a web-based interface and a high-performance computing (HPC) backbone, the user-friendly web-interface allows the users to easily manipulate their data and run complex and computationally heavy tasks without any HPC knowledge. While having already gained a large and steady user-base, the BioPortal is currently undergoing a complete re-implementation in order to even better meet the requirements of the modern research communities. Among the substantial feature enhancement are better support for reproducibility of research procedures and shareability of the scientific data (both input and output), as well a generally more multi-user centric design. The re-implementation of BioPortal is based on the Galaxy framework (Giardine et al., 2005) which we return to below.

A foundational design decision when building a system like LAP is whether to start from scratch or build on existing frameworks. Several frameworks that offer means to combine analysis services into pipelines have surfaced in the last decade, providing APIs for integrating and

[4]Please see `http://www.bioportal.uio.no` for background.
[5]The WebLicht website: `http://weblicht.sfs.uni-tuebingen.de/weblichtwiki/`
[6]For more information in Spanish, please see `http://clarin-es-lab.org/`
[7]The Language Grid website: `http://langrid.org/en/`

developing tools as well as offering an array of pre-installed services. GATE (Cunningham et al., 2011) and UIMA (Götz and Suhre, 2004) are two noteworthy efforts in the language technology realm, while Taverna (Missier et al., 2010) and Galaxy have mostly been used for research in the life sciences. We found Galaxy to be the most suitable first choice for our pilot experiments for the simple reason that its off-the-shelf feature-set seems, for the most part, compatible with our vision for the portal. While GATE, UIMA and Taverna all offer graphical interfaces that allow end-users to combine tools visually, they are required to install desktop applications which, to different degrees, may be difficult to set up due to, for instance, dependencies to other software packages.

Though systems like UIMA, GATE and Taverna can be ported to work within a browser, the Galaxy framework is natively a web-application, offering a full-blown, intuitive interface, eliminating the need to spend developer time on porting applications to the browser. Taverna and Galaxy take a similar approach to visual pipelining, allowing users to draw directed graphs where each node represents a processing tool with inputs and outputs; Galaxy, however, also implements workspaces that allow users to organize files in different groups, called *histories*, where the outputs generated from each annotator in the workflow are collected.

Another reason for adopting Galaxy for LAP is to cooperate and exchange knowledge with the BioPortal developers at USIT, who are currently re-implementing the system on top of Galaxy. This means we can benefit from existing local expertise when it comes to adapting and integrating tools, connecting it to an HPC backbone, communicating with authentication services, and other technical challenges.

Turning again to BioPortal, it is worth mentioning that the decision to adopt Galaxy as a replacement for the existing implementation was partly due to the large Galaxy developer community and the framework's infrastructure. Galaxy also possesses some characteristics which make it very attractive both for the common user and developers / administrators: Firstly, Galaxy is very "collaborative" in that it allows users to share workflows and intermediate or final results of the data processing at any stage of the computation. This sharing procedure is quick and entirely user-driven. Galaxy supports a complex organization of users into groups (roles) with different levels of cross-group access and permissions over the datasets. Moreover, Galaxy not only facilitates the installation of new resources but also the tuning and maintenance of the existing ones through a web-tool middleware layer. This layer renders the administration of the resources fast and easy. Finally, due to the web interface, Galaxy can easily be adapted as a front-end to large computational resource(s), like storage, grids, and cloud services.

4 Creating a LAP Pilot

In this section we describe the details on implementing a preliminary pilot instance of LAP. While the pilot is already operable, this should still be considered as work in progress. Creating a pilot 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. 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.

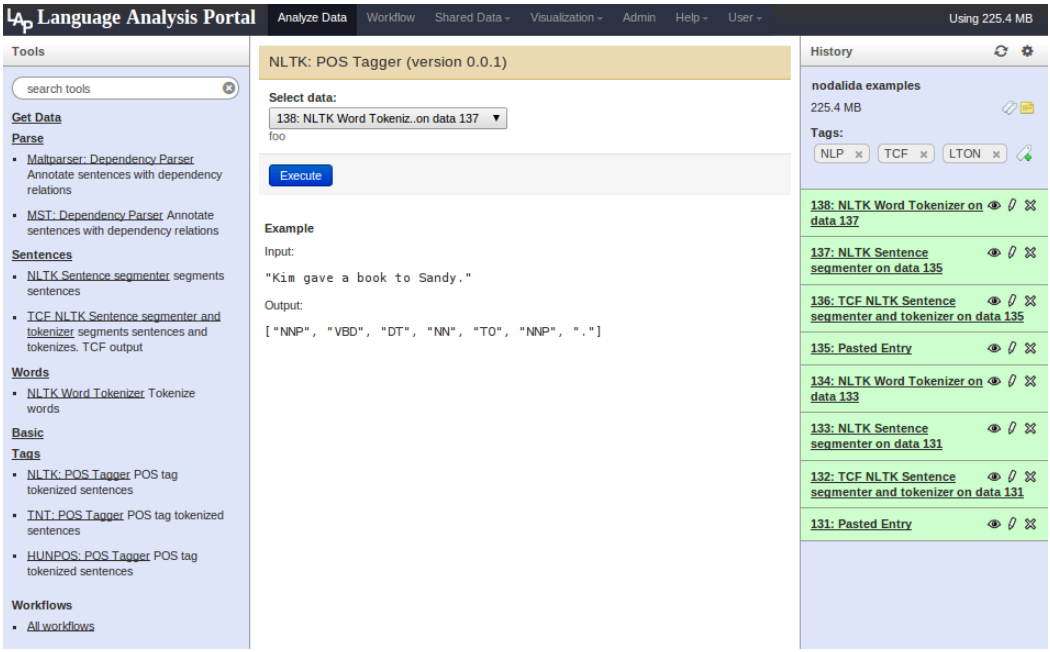

Figure 1: A screenshot of the current, work-in-progress implementation of LAP within Galaxy.

Currently, our engineering efforts are focused on four areas: integrating and adapting a preliminary selection of LT tools, modifying the Galaxy UI, integrating and evaluating interchange formats, and implementing the connection to the HPC cluster (Abel). Below we discuss these issues in turn.

4.1 Galaxy and Workflows

Figure 1 shows the three main panels of the Galaxy UI as adapted for LAP. The left panel contains a list of the installed tools and available workflows; clicking on an item updates the center panel with its details, allowing users to run (or re-run) the tool using one of the elements present in the history panel on the right. Users can create as many histories as they need, using tags and a short description to keep potentially large collections of files in order. The center panel also houses the workflow manager, where processing pipelines such as the one pictured in Figure 2 are designed. Additionally, Galaxy makes histories and workflows searchable, allowing users to share them and collaborate.

While the larger LAP user-group includes researchers and students from the humanities, social sciencies, linguistics and language technology itself, the pilot release of the portal will only focus on the latter. A typical use case for the working language technologist, and one that the LAP pilot will provide the means to accomplish, could involve annotating a large text corpus, e.g., a snapshot of Wikipedia, with syntactic dependencies. Furthermore, the researcher or student in question could be interested in producing annotations generated using different parsers, that are in turn invoked with part of speech tags that originate from different upstream annotators. Such an endeavour would minimally include the following steps: (a) log into the

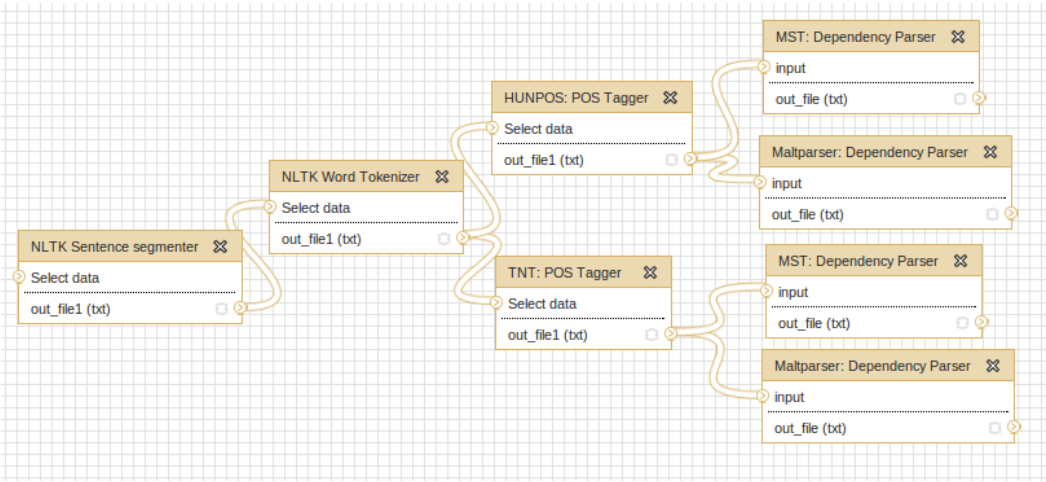

Figure 2: A LAP workflow with four endpoints.

LAP workspace; (b) Create a *history* for the experiment; (c) design and save the workflow, like the example given in Figure 2; (d) run the workflow and (e) download the output files when completion is notified (either on-screen or via email).

4.2 Interchange formats

In order make a heterogeneous set of language processing tools interoperable, datasets have to be converted to and from the required tool-specific representations at each step of the processing. In this context, interchange formats work as a kind of 'trade language' between the tools in the chain. LAP aims to be compatible with other CLARIN-related projects in terms of interchange formats, and provide tools that enable converting to and from widely adopted representations, like the tab-separated CoNLL 2007 format or Penn Tree Bank-style phrase-structure trees.

For LAP's system-internal representation, we are at the moment looking into both TCF (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. Figure 3 shows an example of both representations after running the NLTK (Bird et al., 2009) sentence segmenter and tokenizer on the text "*Sandy barks. Kim Snores.*". In TCF, tokens are the smallest unit in the representation and serve as anchor points for downstream processing, while in LTON corpora are annotated according to a notion of *annotation levels* (e.g., sentence, paragraph, document and so on), with lower-level annotations being encapsulated within higher ones. Finding a suitable internal representation is by no means a trivial task, especially given that relevant datasets may potentially be very large, and rapidly increase in size and complexity even further due to annotations accumulating through involved workflows. Evaluation and integration of these two interchange formats is currently ongoing and a full LAP-implementation of both is expected in time for the pilot release.

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. Additionally, the

TCF

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<?xml-model href="http://de.clarin.eu/images/weblicht-
       tutorials/resources/tcf-04/schemas/latest/d-spin_0_4.
       rnc" type="application/relax-ng-compact-syntax"?>
<ns5:D-Spin xmlns="http://www.dspin.de/data/metadata" xmlns:
       ns2="http://www.dspin.de/data/extdata" xmlns:ns3="http
       ://www.dspin.de/data/textcorpus" xmlns:ns4="http://www.
       dspin.de/data/lexicon" xmlns:ns5="http://www.dspin.de/
       data" version="0.4">
   <MetaData/>
   <ns3:TextCorpus lang="en">
       <ns3:text>Sandy barks. Kim snores.</ns3:text>
       <ns3:tokens>
           <ns3:token ID="t_0">Sandy</ns3:token>
           <ns3:token ID="t_1">barks</ns3:token>
           <ns3:token ID="t_2">.</ns3:token>
           <ns3:token ID="t_3">Kim</ns3:token>
           <ns3:token ID="t_4">snores</ns3:token>
           <ns3:token ID="t_5">.</ns3:token>
       </ns3:tokens>
       <ns3:sentences>
           <ns3:sentence tokenIDs="t_0 t_1 t_2"/>
           <ns3:sentence tokenIDs="t_3 t_4 t_5"/>
       </ns3:sentences>
   </ns3:TextCorpus>
</ns5:D-Spin>
```

LTON

```
{
    "annotations": [
        {
            "annotators": {
                "token:nltk": [
                    "Sandy",
                    "barks",
                    "."
                ]
            },
            "text": "Sandy barks."
        },
        {
            "annotators": {
                "token:nltk": [
                    "Kim",
                    "snores",
                    "."
                ]
            },
            "text": "Kim snores.\n"
        }
    ],
    "level": "sentence:nltk",
    "name": "/lap-galaxy/database/files/000/dataset_138.dat"
}
```

Figure 3: An example of a toy corpus annotated with sentences and tokens, formatted in the two candidate interchange formats for LAP.

wrapper handles the submission of the job to the Abel cluster.

4.3 The Abel HPC Cluster

Abel[8] is the name of the high performance computing facility at UiO, hosted by the USIT Research Computing group. The powerful Linux cluster is a shared resource for research computing, boasting more than 600 machines, totaling more than 10.000 cores (CPUs). At the time of writing, the cluster ranks at position 134 on the list of top 500 supercomputers worldwide.[9] Among its frequent users, besides the language technology group, we find researchers from the life sciences, astrophysics, geophysics, and chemistry.

When executing a workflow from the LAP instance of Galaxy, each component task will in turn be submitted to the job queue on the Abel cluster. Control is then temporarily delegated to the cluster queue system—using a job scheduler and resource manager called SLURM[10]—before the produced output is finally returned to Galaxy. An important part of the work on adapting Galaxy for LAP (and the BioPortal) is to make this connection as seamless as possible.

For the pilot release, LAP's toolshed will provide enough language processing tools to enable a user test session involving master students of the language technology program at UiO. The inventory will include typical annotators such as sentence segmenters, tokenizers, lemmatizers, chunkers and syntactic parsers; these will, to a varying degree, be configurable in terms of e.g., models, tagsets and syntactic paradigms. In terms of language coverage, the pilot will provide tools for processing Scandinavian languages in addition to English. The test session, which is planned for early Q2 2013, will pave the way for further development.

[8] For more detailed information on the Abel computing cluster, see `http://uio.no/hpc/abel/`.

[9] For more information about the ranking, please see `http://www.top500.org`.

[10] Simple Linux Utility for Resource Management

5 Conclusion and Outlook

This paper has laid out the long-term goals and plans for creating an online portal for language analysis (LAP). The effort forms part of the CLARINO infrastructure project and one of the overall goals is to make language technology readily accessible and usable for researchers from the humanities and social sciences. With easy access through Feide (and CLARIN AAI) authentication, the web portal will provide a uniform graphical interface to a large repository of LT tools installed on a high-performance computing cluster. Users will be able to create complex workflows using an intuitive interface, and each user will have access to a personal workspace for storing data persistently across sessions. In terms of linguistic coverage, LAP will focus on languages most actively used in Norway, e.g., Norwegian *Bokmål* and *Nynorsk*, Sámi, other Scandinavian languages, and English.

The development of the portal is still in its early stages, and in addition to presenting the overall vision of LAP in its final state, this paper has documented the work towards an already functioning pilot. This preliminary version of the portal will act as a proof-of-concept, helping to inform strategic decisions about the remaining work, as well as a demonstrator that can be used when reaching out to tool developers and when surveying user needs.

One of the core components of the current pilot version of the portal is the Galaxy workflow manager—a web-based system initially developed for data intensive research in the biomedical domain. Galaxy is already deployed in a portal for bioinformatics research, BioPortal, also hosted at the University of Oslo. This means we can benefit from existing local expertise when it comes to adapting and integrating tools in Galaxy, connecting it to an HPC backbone, and other technical challenges.

The pilot release of the system will address the requirements of users with a language-technological background, with a closed user-test session planned for early Q2 and an open pilot release in Q3 2013. Further work will address the challenge of shaping LAP into a useful research tool for the humanities and the social sciences, investigating possible use-cases and collecting user-requirements from active researchers from these fields.

References

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly.

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.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M. A., Saggion, H., Petrak, J., Li, Y., and Peters, W. (2011). *Text Processing with GATE (Version 6)*.

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.

Götz, T. and Suhre, O. (2004). Design and implementation of the UIMA common analysis system. *IBM Syst. J.*, 43(3):476–489.

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.

Missier, P., Soiland-Reyes, S., Owen, S., Tan, W., Nenadic, A., Dunlop, I., Williams, A., Oinn, T., and Goble, C. (2010). Taverna, reloaded. In *Proceedings of the 22nd International Conference on Scientific and Statistical Database Management*, pages 471–481.