LAP: The CLARINO Language Analysis Portal

Emanuele Lapponi, Stephan Oepen, Arne Skjærholt, and Erik Velldal
University of Oslo
Department of Informatics
{emanuel|oe|arnskj|erikve}@ifi.uio.no

1 Introduction: High-Level Goals

This abstract describes the current state of the Language Analysis Portal (LAP) currently under development in the Norwegian CLARINO initiative. LAP provides users with a collection of state-of-the-art tools for natural language processing that are accessible via a unified, in-browser user interface. Built on top of the open-source Galaxy framework (Giardine et al., 2005; Blankenberg et al., 2010; Goecks et al., 2010), the system offers means to combine tools into workflows, keep track of their output, and deliver results to users in different formats.

Unlike related on-line processing environments such as Weblicht (Hinrichs et al., 2010), LAPPS (Ide et al., 2014) and Alveo (Estival and Cassidy, 2014), which predominantly instantiate a distributed architecture of web services, LAP achieves scalability to potentially very large data volumes through integration with the Norwegian national e-Infrastructure, and in particular job submission to a capacity compute cluster. This setup leads to tighter integration requirements and also calls for efficient, low-overhead communication of (intermediate) processing results with workflows. We meet these demands by coupling the data model of the Linguistic Annotation Framework (LAF) (Ide and Romary, 2001; Ide and Suderman, 2013) with a lean, non-redundant JSON-based interchange format and integration through an agile and performant NoSQL database—allowing parallel access from cluster nodes—as the central repository of linguistic annotation. While the utility of natural language processing tools is apparent for linguists (computational or not), the ultimate goal of LAP is to reduce technological barriers to entry for researchers from the social sciences and humanities (SSH).

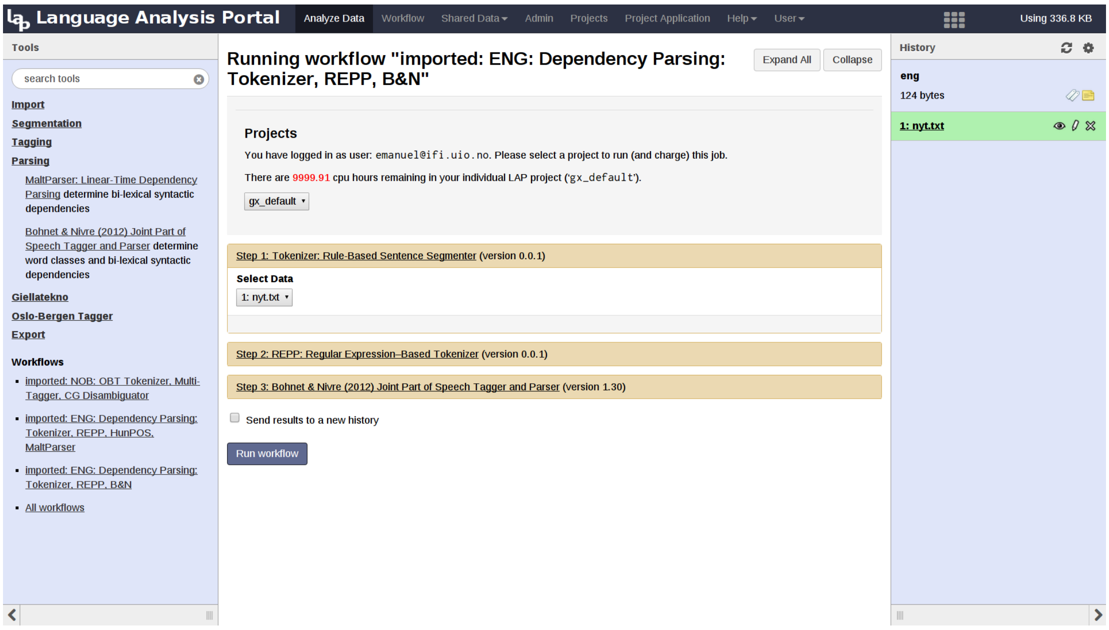
2 Design and Implementation: Galaxy and HPC

As described in Lapponi et al. (2013), LAP is built on top of Galaxy, a widely adopted framework for genome processing in the field of bioinformatics. Galaxy has proven to also be a suitable platform for natural language processing portals, and it has recently also been adopted by e.g. LAPPS[1] and Alveo.[2] Galaxy is an application that runs inside the browser, offering a graphical user interface to configure and combine analysis tools, upload, inspect and download data and share results and experiments with other users. A central part of the interface is 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.

Rather than creating *ad-hoc* processing tools for LAP, existing NLP software is adapted and made available from within Galaxy. Adapting a tool to exist within the LAP ecosystem means making sure that it is compatible with the other installed tools: For instance, a dependency parser that requires sentence

[1] http://galaxy.lappsgrid.org/

[2] http://alveo.edu.au/help/analysing-data/transferring-data-to-galaxy-for-processing/



Running workflow "imported: ENG: Dependency Parsing: Tokenizer, REPP, B&N"

MaltParser: Linear-Time Dependency Parsing determine bi-lexical syntactic dependencies

Bohnet & Nivre (2012) Joint Part of Speech Tagger and Parser determine word classes and bi-lexical syntactic dependencies

- imported: NOB: OBT Tokenizer, Multi-Tagger, CG Disambiguator
- imported: ENG: Dependency Parsing: Tokenizer, REPP, HunPOS, MaltParser
- imported: ENG: Dependency Parsing: Tokenizer, REPP, B&N
- All workflows

Projects

You have logged in as user: emanuel@ifi.uio.no. Please select a project to run (and charge) this job.

There are 9999.91 cpu hours remaining in your individual LAP project ('gx_default').

gx_default

Step 1: Tokenizer: Rule-Based Sentence Segmenter (version 0.0.1)

Select Data: 1: nyt.txt

Step 2: REPP: Regular Expression–Based Tokenizer (version 0.0.1)

Step 3: Bohnet & Nivre (2012) Joint Part of Speech Tagger and Parser (version 1.30)

☐ Send results to a new history

eng
124 bytes

1: nyt.txt

Figure 1: Screenshot of the LAP user interface: Preparing to launch a pre-defined workflow.

segmentation, tokenization, and part-of-speech annotations should be able to build its input from the output of other LAP annotators.

As different tools often use different and mutually incompatible representation formats for encoding input and output data, an important step in achieving interoperability is the implementation of an interchange format that can serve as a *lingua franca* among the components in a workflow. In LAP this interchange format is based on the LAF data model for representing annotations, as further described below.

3 Tool Interchange Format: LAF in LAP

LAF is a graph-based model for representing multi-modal linguistic annotations that aims at providing full interoperability among annotation formats. One of the fundamental principles that guided the development of LAF is that all annotation information should be explicitly represented, i.e., the interpretation of annotations should not require implicit knowledge about particular categories and relations (Ide and Suderman, 2013). Another fundamental principle is that one should observe a strict separation between annotation *structure* and annotation *content*. The focus of LAF is only on the structural part, and it is important to realize that LAF itself is not a format as such. It does not come with pre-specified linguistic labels or categories etc. Rather, it is a general framework for how to represent the annotation structure itself; an abstract data model specifying how to relate annotations to data and how to relate annotations to other annotations.

In LAP, annotations produced by tools are tied to both the original text and each other by means of the main components in a LAF graph: regions, nodes and edges. Regions describe the so-called base segmentation of a text in terms of character offsets, and might be associated to nodes containing annotation about the segment. For instance a tokenizer, which accepts free text as input, can produce both a target region and a paired node with an annotation containing the normalized token. A part-of-speech tagger produces both nodes containing the actual POS annotation and edges linking its nodes to the input tokens. Region, node and edge records are added to a MongoDB instance after each tool execution.

When the user runs another tool, only the subgraph containing the necessary annotations is invoked. For instance, if the user wants to run a new POS tagger on text that has already been tagged and parsed, LAP invokes only the part of the graph describing the relevant tokens. This strategy differs from other approaches adopted in other web interfaces to NLP tools. In Weblicht, for instance, annotations in the

TCF format (Heid et al., 2010) have to be parsed and re-encoded at each processing step. We believe that our approach is better suited to the modularity of LAP and provides a more malleable set-up for efficiently analyzing larger data-sets and running out-branching jobs on the cluster.

However, it is important to clarify that our implementation of the LAF data model, due to fundamental differences in goals and setup, does not stand in competition with TCF or richer formats for corpus annotations such as FoLiA (van Gompel and Reynaert, 2013) or TEI as an end-user format. The focus of our LAF implementation is its use as an internal interchange format. We have no expectations that future LAP users will find much use in downloading LAP annotations directly from the database. In keeping with the modular philosophy of LAP, we aim to provide users with LAP tools for importing from and exporting to other formats (imagine, for instance, a workflow starting with a TCF importer and ending in a TCF exporter, enabling compatibility with Weblicht workflows, or exporting to the FoLiA format for further manual annotations with the FLAT[3] annotation tool). A detailed description of the LAF data model as implemented in LAP is provided by Lapponi et al. (2014).

4 Tool Integration and Versioning: The LAP Tree

LAP integrates processing tools from many different sources, which are implemented in different programming languages. We have designed and populated a repository of tools, i.e. ready-to-run and pre-configured installations of processing components used by LAP, with replicability and relocatability as key design desiderata; this repository is dubbed the *LAP Tree*. Central installation of static versions of LAP tools on compute nodes is completely avoided. Instead, the LAP Tree is realized as a version-controlled repository, where individual components and all dependencies that transcend basic operating system functionality can (in principle) be checked out by an arbitrary user and into an arbitrary location, for immediate use (on all cluster nodes) through Galaxy. By default, Galaxy users are presented with the latest (stable) available version of tools, but the LAP Tree makes it possible for users to request individual versions of components directly in Galaxy, which 'behind the scenes' is implemented by means of a user-specific instance of (relevant parts of) the LAP Tree that is dynamically populated.

All code at the LAP interface layer is versioned jointly with the LAP Tree, such that we anticipate enabling users to back-date workflows to (in principle) arbitrary points in time of LAP evolution, thus taking an important step to reproducibility of experiments and replicability of results. Galaxy provides built-in functionality for sharing data with other users, including processing results, as well as complete workflows. In this context, a specific, per-user instantiation of a processing workflow can encompass all parameter choices (including tool versions) for all tools involved, i.e. the full 'recipe' that lead from a set of input data to a set of results. Once frozen as a workflow and shared with other users, the combination of versioning in the LAP Tree and standard Galaxy functionality, thus, promises to provide a framework that enables reproducibility (and, ultimately, also adaptation) by other researchers.

5 Reaching Out: Analyzing the European Parliament Debates

An important part of the motivation for CLARIN(O) as an infrastructure initiative is, of course, to facilitate the use of language technology among SSH researchers. This is also an important motivation for LAP. In attempting to facilitate and enable LT-based research in other fields, the importance of maintaining a bottom-up view on the process of how research questions are created and addressed in practice should not be underplayed (Zundert, 2012). Our position is that starting out from actual research questions, in direct collaboration with SSH researchers themselves, provides an ideal point of departure for surveying user requirements. In line with this we have focused on establishing contact with SSH researchers that might be interested in collaborating on using language technology in their own work. One such outreach effort has led to collaboration with researchers within political science interested in data-driven analysis of the legislative processes within the European Union. A joint ongoing project investigates whether an SVM classifier can be trained to predict the party affiliations of Members of the European Parliament on the basis of their speeches in the plenary debates, with a particular focus on the contribution of linguistically informed features. Preliminary results are presented by Høyland et al.

[3] https://github.com/proycon/flat/

(2014). The stages in the prediction pipeline here involve several layers of linguistic annotations, in addition to feature extraction, model estimation, and testing. An important future step is to fully integrate this entire pipeline within LAP itself, seeking to find the right balance between supporting all the requirements of this particular analysis task while also maintaining enough generality in the implementation to also make the involved components re-usable and applicable to other tasks. The work described by Høyland et al. (2014) ties in closely with the focus of the CLARIN-initiaited project *Talk of Europe: Travelling CLARIN Campus*[4] (ToE), focusing on processing and representing the European Parliament debates in a way that enables further analysis and collaboration by SSH researchers).

6 Talk of Europe: LAP Annotations in a Large RDF Store

The Talk of Europe project aims at curating the proceedings of the European Parliament Debates to linked data, so that it can be linked to and reused by other datasets and services. Our LAP-specific objective in this context is to contribute to the creation of a high- quality, multi-lingual corpus of European Parliament Proceedings, coupled with state-of-the-art syntactico-semantic analyses at the token and sentence levels—all encoded in the form of labeled directed graphs in the Resource Description Framework (RDF). LAP developers participated in the first ToE Creative Camp, where our contribution has targeted the textual (transcribed) content of European Parliament speeches. Leveraging in-house experience and technology for syntactic and semantic parsing of running text, primarily in English, we have worked with ToE colleagues at *Vrije Universiteit Amsterdam* to (a) help improve the content extraction pipeline and (b) establish consensus of a ToE-compatible RDF encoding of LAP annotations. We expect that making available a standardized collection of (automatically acquired) linguistic annotations of the raw text in the corpus, related to the resource at large through RDF links, will enable a variety of downstream analytical tasks and aid replicability and comparability of results.

The LAP workflow that produces these annotations consists of the CIS Tokenizer sentence segmenter, the REPP word tokenizer (Dridan and Oepen, 2012), and the Bonhet and Nivre part-of-speech tagger and dependency parser (Bohnet and Nivre, 2012). The resulting annotations are then exported to RDF triples as defined by a LAP ontology, and are linked to the relevant speech using named graphs in TriG syntax. As of September 2015, the ToE data consists of 292,379 speeches, amounting to roughly 63 million tokens and resulting in approximately 2.7 billion triples once annotated in LAP. An example showing both the form of the annotations and how to access them via a SPARQL endpoint can be found on the LAP development wiki pages. [5] Integration of this vast collection of LAP-derived triples with the main ToE store at Amsterdam remains to be negotiated.

7 Current State of Development and Future Plans

A LAP development instance has been available for trial use since late 2014,[6] with a production instance set to launch towards the end 2015. The tools currently installed in LAP target Norwegian and Sami as well as English, allowing users to process raw text, analyze it, and export the resulting annotations in different formats: various tabulator-separated formats in the tradition of the shared tasks of the Conference on Natural Language Learning; two common variants of the Constraint Grammar textual exchange format; and the RDF representation designed for the ToE use case. Documentation guiding new users through their first sessions in LAP is available in the LAP project pages.[7] Our short- to mid-term goals with regard to tool development include (a) broadening the range of processing types covered (for example to support language identification; ‘deeper’, semantic parsing; training and application of document- and substring-level classifiers; and others), (b) supporting parameterizable extraction of (relevant) linguistic content from various mark-up formats, as well as (c) developing import and export interfaces with other CLARINO platforms such as the Corpuscle and Glossa corpus search services.

[4] http://www.talkofeurope.eu/

[5] http://moin.delph-in.net/LapDevelopment/ToE

[6] https://lap.hpc.uio.no/

[7] http://www.mn.uio.no/ifi/english/research/projects/clarino/user/

The LAP trial instance currently supports authentication through the Norwegian national Feide federation and the world-wide eduGAIN interfederation, which in in late 2015 counts some 40 national federations among its members. In addition to the above, the forthcoming production instance will also allow authorization via the CLARIN IdP and Feide OpenIdP services (enabling new users to self-register), as it appears that some CLARIN member sites regrettably have yet to gain access to eduGAIN. Use of the Norwegian national supercomputing e-infrastructure 'behind the scenes' of LAP mandates full accountability, where the individual allocation of compute cycles will depend on user affiliations and the choice of authentication service. At present, (a) individual users (including international ones) can be granted a standard quota of up to 5,000 cpu hours per six-month period; (b) users or groups of users can apply for LAP projects, where a project typically can receive an allocation of up to 100,000 cpu hours per period; and (c) users or groups of users can independently acquire separate allocations through the Norwegian national Notur services, which they can transparently use for LAP computation.

For compatibility with other Galaxy-based portals operated at the University of Oslo (UiO) and in the national ELIXIR.NO network, the LAP trial instance currently still builds on a 2013 snapshot of the Galaxy code base, but the forthcoming production instance will be the first UiO portal running on a current (March 2015) Galaxy release. While the Galaxy framework continues to evolve dynamically, the so-called 'tool descriptions'—(mostly) declarative specifications of the interfaces and customization options for a specific processing tool—are often not fully portable across different Galaxy versions. With the launch of the LAP production instance, we expect to emphasize reproducibility and replicability (see §4 above). While the LAP Tree and API to the annotation database were designed with these goals in mind, the interface to Galaxy proper may, thus, pose technical challenges when future LAP instances migrate to newer Galaxy code; we optimistically expect that the tool description framework will evolve in a backwards-compatible manner, but in the extreme changes at this interface level might call for updates to not only current revisions of tool descriptions but also to historic ones.

As our choice of building the CLARINO Language Analysis Portal on Galaxy has recently been followed by the US-based LAPPS and Australian Alveo initiatives (see above), we plan to reach out to these developer communities and exchange experiences (as well as strengthen the recognition of NLP-specific needs in the Galaxy community), for example through a jointly organized workshop at the 2016 Galaxy Community Conference.

References

Daniel Blankenberg, Gregory Von Kuster, Nathaniel Coraor, Guruprasad Ananda, Ross Lazarus, Mary Mangan, Anton Nekrutenko, and James Taylor. 2010. Galaxy. A web-based genome analysis tool for experimentalists. *Current Protocols in Molecular Biology*, page 19.10.1 – 21, January.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Conference on Natural Language Learning*, page 1455 – 1465, Jeju Island, Korea.

Rebecca Dridan and Stephan Oepen. 2012. Tokenization. Returning to a long solved problem. A survey, contrastive experiment, recommendations, and toolkit. In *Proceedings of the 50th Meeting of the Association for Computational Linguistics*, page 378 – 382, Jeju, Republic of Korea, July.

Dominique Estival and Steve Cassidy. 2014. Alveo, a human communication science virtual laboratory. In *Australasian Language Technology Association Workshop 2014*, page 104.

Belinda Giardine, Cathy Riemer, Ross C. Hardison, Richard Burhans, Laura Elnitski, Prachi Shah, Yi Zhang, Daniel Blankenberg, Istvan Albert, James Taylor, Webb Miller, W. James Kent, and Anton Nekrutenko. 2005. Galaxy. A platform for interactive large-scale genome analysis. *Genome Research*, 15(10):1451 – 5, October.

Jeremy Goecks, Anton Nekrutenko, James Taylor, and The Galaxy Team. 2010. Galaxy. A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology*, 11(8:R86), August.

Ulrich Heid, Helmut Schmid, Kerstin Eckart, and Erhard W. Hinrichs. 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*, page 494 – 499, Valletta, Malta.

Marie Hinrichs, Thomas Zastrow, and Erhard W Hinrichs. 2010. Weblicht: Web-based lrt services in a distributed escience infrastructure. In *LREC*.

Bjørn Høyland, Jean-François Godbout, Emanuele Lapponi, and Erik Velldal. 2014. Predicting party affiliations from European Parliament debates. In *Proceedings of the 52nd Meeting of the Association for Computational Linguistics: Workshop on Language Technologies and Computational Social Science*, page 56 – 60, Baltimore, MD, USA.

Nancy Ide and Laurent Romary. 2001. A common framework for syntactic annotation. In *Proceedings of the 39th Meeting of the Association for Computational Linguistics*, page 306 – 313, Toulouse, France, July.

Nancy Ide and Keith Suderman. 2013. The Linguistic Annotation Framework: A standard for annotation interchange and merging. *Language Resources and Evaluation*, (forthcoming).

Nancy Ide, James Pustejovsky, Christopher Cieri, Eric Nyberg, Denise DiPersio, Chunqi Shi, Keith Suderman, Marc Verhagen, Di Wang, and Jonathan Wright. 2014. The language application grid. *Proceedings of the Ninth International Language Resources and Evaluation (LREC14), Reykjavik, Iceland. European Language Resources Association (ELRA)*.

Emanuele Lapponi, Erik Velldal, Nikolay A. Vazov, and Stephan Oepen. 2013. Towards large-scale language analysis in the cloud. In *Proceedings of the 19th Nordic Conference of Computational Linguistics: Workshop on Nordic Language Research Infrastructure*, page 1 – 10, Oslo, Norway.

Emanuele Lapponi, Erik Velldal, Stephan Oepen, and Rune Lain Knudsen. 2014. Off-road LAF: Encoding and processing annotations in NLP workflows. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, page 4578 – 4583, Reykjavik, Iceland.

Maarten van Gompel and Martin Reynaert. 2013. FoLiA: A practical xml format for linguistic annotation - a descriptive and comparative study. *Computational Linguistics in the Netherlands Journal*, 3:63–81, 12/2013.

Joris van Zundert. 2012. If you build it, will we come? Large scale digital infrastructures as a dead end for digital humanities. *Historical Social Research*, 37(3).