# A Workflow for Integrating Close Reading and Automated Text Annotation

Maciej Janicki[1], Eetu Mäkelä[1], Anu Koivunen[2], Antti Kanner[1], Auli Harju[2], Julius Hokkanen[2], and Olli Seuri[3]

[1] Department of Digital Humanities, University of Helsinki
[2] Faculty of Social Sciences, Tampere University
[3] Faculty of Information Technology and Communication Studies, Tampere University

**Abstract.** We present a workflow for Digital Humanities projects allowing to combine close and distant reading, as well as automated text annotation, in an iterative process. We rely on mature tools and technologies, like R, WebAnno or Prolog, that are combined in a highly automated pipeline. Such architecture can deal well with underspecified and frequently changing requirements and allow a continuous exchange of information between the computational and domain experts in all stages of the project. The workflow description is illustrated on a concrete example concerning news media analysis.

**Keywords:** Workflow, Close Reading, Distant Reading, Interdisciplinary Cooperation, CSV.

## 1 Motivation

Digital Humanities projects often involve application of language technology or machine learning methods in order to identify phenomena of interest in large collections of text. However, in order to maintain credibility for humanities and social sciences, the results gained this way need to be interpretable and investigable and cannot be detached from the more traditional methodologies which rely on close reading and real text comprehension by domain experts. The bridging of those two approaches with suitable tools and data formats, in a way that allows a flow of information in both directions, often presents a practical challenge.

In our research, we have developed an approach to digital humanities research that allows combining computational analysis with the knowledge of domain experts in all steps of the process, from the development of computational indicators to final analysis [4]. Put succintly, our approach hinges on, as early as possible, creating an environment where both the research data as well any computational enrichments and analyses done on it can be shown, pointed to and discussed, both from the perspective of the domain experts as well as from the perspective of the computational experts. Further, because at the start of the project neither the computational indicators nor axes of analysis are yet finalized, the environment must support easy iterative updating.

In this poster, we describe a particular implementation of this approach, as it appears in the project: *Flows of Power: media as site and agent of politics*. This project is a collaboration between journalism scholars, linguists and computer scientists aimed at the analysis of the political reporting in Finnish news media over the last two decades. We study both the linguistic means that media use to achieve certain goals (like appearing objective and credible, or appealing to the reader's emotions), as well as the structure of the public debate reflected there (what actors get a chance to speak and how they are presented). What we will here be particularly focusing on are the technical aspects, as they relate to 1) enabling interaction between different elements of our development and analysis environment and 2) enabling iterative development.

## 2 Software and Data Formats

As many research questions in our project concern linguistic phenomena, a Natural Language Processing pipeline is highly useful. We employ the Turku neural parser pipeline [2], which provides dependency parsing, along with lower levels of annotation (tokenization, sentence splitting, lemmatization and tagging). Further, we apply the rule-based FINER tool [5] for named entity recognition.

*R and Shiny.* Our primary toolbox for statistical analysis is R. This motivates using the 'tidy data' CSV format [6] as our main data format. In order to keep the number and order of columns constant and predictable, only the results of the dependency parsing pipeline are stored together with the text, in a one-token-per-line format very similar to CONLL-U.[4] All additional annotation layers, beginning with named entity recognition, are relegated to separate CSV files, where tuples like (*documentId*, *sentenceId*, *spanStartId*, *spanEndId*, *value*) are stored. Such tabular data are easy to manipulate within R.

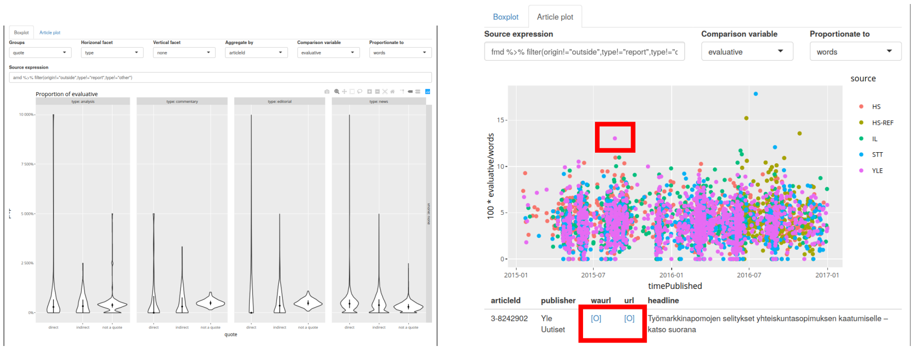
In terms of applications and interfaces, we favour web applications over locally installed ones. First, these have a lower barrier of entry, being available for use from anywhere a web browser is installed. Second, they allow easier sharing of views with other project participants through copying and pasting of stable URLs. This is important, as in our approach to the research process, the focused sharing of examples of both in-domain objects of interest, as well as the results of automated processing plays a crucial part in building a common understanding, and guiding development and analysis. Finally, because our work requires iterative development, it is much easier to update both data as well as functionality centrally once, as opposed to everyone needing to download newer versions of data and programs all the time.

Based on the above considerations, for distant reading and discovery of statistical patterns, we rely on a Shiny[5] Web application that we developed ourselves (Fig. 1). It allows easy access to aggregate views of the dataset based on variables like the proportion of quotes, affective expressions, or other automatically generated annotations. The scatterplot view (Fig. 1, right) is useful for drilling down into examples of various parts of the distributions, and particularly for detecting and exploring outliers. In this view, each point represents an article, and clicking on it shows detailed information, including the headline, source and a link to our close reading interface (see below). This functionality is illustrated in the figure by the red frames: the upper one shows the

[4] `https://universaldependencies.org/format.html`

[5] Shiny is a framework for building Web-based user interfaces in R.

point that has been clicked, and the lower one the details of that article. Through this, researchers are both able to interpret what particular portions of the distribution mean in practice in the texts they represent, as well as examine whether outliers are caused by errors in our processing pipeline, or real in-domain phenomena of interest.



Groups: quote
Horizonal facet: type
Vertical facet: none
Aggregate by: articleId
Comparison variable: evaluative
Proportionate to: words
Source expression: amd %>% filter(origin!="outside",type!="report",type!="other")

Source expression: fmd %>% filter(origin!="outside",type!="report",type!="c
Comparison variable: evaluative
Proportionate to: words

| articleId | publisher | waurl | url | headline |
|---|---|---|---|---|
| 3-8242902 | Yle Uutiset | [O] | [O] | Työmarkkinapomojen selitykset yhteiskuntasopimuksen kaatumiselle – katso suorana |

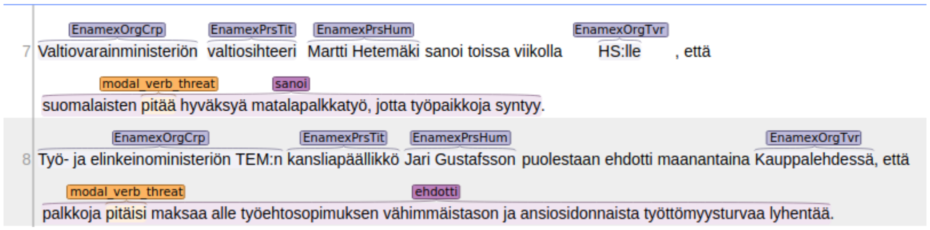
**Fig. 1.** The Shiny app for explorative analysis of statistical patterns.

*WebAnno.* For the visualization of automatic annotations, close reading and manual annotation, we decided to employ WebAnno [1].[6] While this tool was originally intended for the creation of datasets for language technology tasks, its functionality is designed to be very general, which enabled its use in a wide variety of projects involving text annotation.[7] In addition to the usual linguistic layers of annotation, like lemma or head, it allows the creation of custom layers and feature sets. WebAnno has a simple but powerful visualization facility: annotations are shown as highlighted text spans, feature values as colorful bubbles over the text, and the various annotation layers can be shown or hidden at user's demand (Fig. 2). This kind of visualization does not disturb close reading. It allows to concentrate on the features that are currently of interest, while retaining the possibility to look into the whole range of available annotations.

WebAnno supports several data formats for import and export. All of them assume one document per file. Among others, different variants of the CONLL format are supported. WebAnno-TSV is an own tab-separated text format, which, as opposed to CONLL, includes support for custom annotation layers. Because it is a text format and is well documented, we were able to implement a fully automatic bidirectional conversion between our corpus-wide, per-annotation CSV files and per-document WebAnno-TSV files. Thus, using WebAnno as an interface to interact with the domain experts who perform close reading and manual annotation, we are able to exchange our results quickly and with a high degree of automatization.

One problem we initially had with integrating WebAnno as part of our ecosystem was that while WebAnno did allow linking to each document by URLs, these were

[6] `https://webanno.github.io/webanno/`
[7] see: `https://webanno.github.io/webanno/use-case-gallery/`



7 Valtiovarainministeriön valtiosihteeri Martti Hetemäki sanoi toissa viikolla HS:lle , että suomalaisten pitää hyväksyä matalapalkkatyö, jotta työpaikkoja syntyy.

8 Työ- ja elinkeinoministeriön TEM:n kansliapäällikkö Jari Gustafsson puolestaan ehdotti maanantaina Kauppalehdessä, että palkkoja pitäisi maksaa alle työehtosopimuksen vähimmäistason ja ansiosidonnaista työttömyysturvaa lyhentää.

**Fig. 2.** WebAnno displaying the automatically obtained annotation layers 'named entity' (grayish blue), 'hedging/threat' (orange) and 'indirect quote' (purple).

based on internal IDs that changed each time we reloaded the data (which, in our iterative development process, happened frequently). In order to increase the interoperability of WebAnno with our other tools, we contributed patches that allow the projects and documents to be referenced by name instead of these internal IDs. Thus, a document `doc` within the project `project` can be accessed via the URL:

http://*webanno-instance-url*/annotation.html?#!pn=*project*&n=*doc*.
This way, the WebAnno view of an annotated document can be easily linked to from any other tool just by knowing the document name.

*Prolog.* Finally, some automatic annotations are produced by rule-based approaches implemented in Prolog. Thus, another document representation that we utilize is a set of Prolog predicates encoding the sentence structure and the linguistic annotation. A schema illustrating the complete back-end with all employed data formats is shown in Fig. 3.

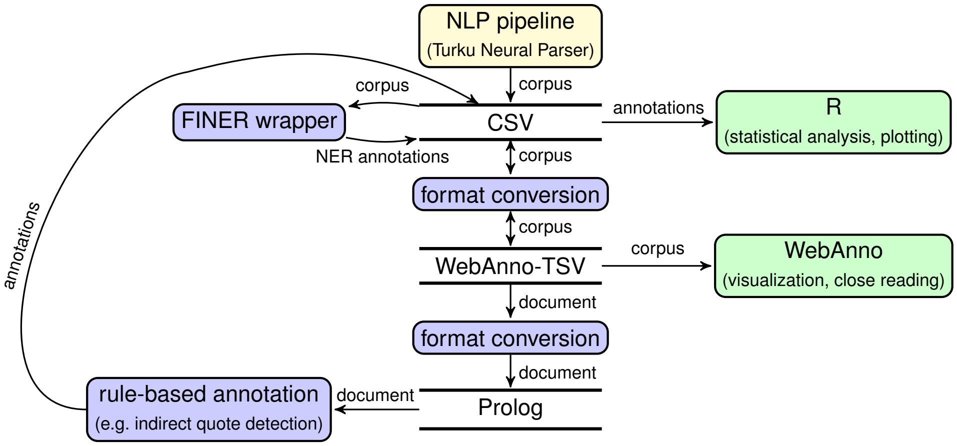


**Fig. 3.** A dataflow diagram of the back-end.

## 3 Case Study: Affective and Metaphorical Expressions in Political News

We applied the methodology outlined above in a recently conducted case study. The subject of the study was the use of affective and metaphorical language in a media debate about a controversial labour market policy reform, called ‘competitiveness pact’ which was debated in Finland in 2015-16.

The linguistic phenomenon in question is complex and not readily defined. It is also context-dependent: ‘the ball is in play’ is metaphoric when referred to politics, but not when referred to sports. There is no straightforward method or tool for automatic recognition of such phrases. Therefore, we started the study with a close reading phase, in which the media scholars identified and marked the phrases they recognized as affective or metaphorical in the material covering the competitiveness pact. The marked passages were subsequently manually post-processed to extract single words with ‘metaphoric’ or ‘affective’ charge. The list of words obtained this way was further expanded with their synonyms, obtained via word embeddings. Using this list, we were able to mark the potential metaphoric expressions in the unread text as well. The results of this analysis were published in [3].

## 4 Conclusion

We have presented a particular realization of our general approach for combining qualitative and quantitative analysis in a cooperation between computational and social sciences. The main characteristic of this approach is utilizing mature existing tools, like R, WebAnno, Prolog and Turku neural parser for specialized subtasks, while focusing our own contributions on a pipeline that combines these tools into an interlinked ecosystem with support for iterative development and focused discussion between the computer scientists and the domain experts. We find such an architecture to be more appropriate than custom-built monolithic environments, as the requirements on the computational toolkit are not known in advance and may frequently change in result of the interaction with the data. The high degree of automatization allows us to rerun the data conversion steps frequently, thus allowing the insights gained via close reading to feed back to automatic annotation and statistical analysis. The usability of this workflow was confirmed in an independently published case study.

## References

1. Richard Eckart de Castilho, Éva Mújdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84, Osaka, Japan, 2016.
2. Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. Turku neural parser pipeline: An end-to-end system for the CoNLL 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 133–142, Brussels, Belgium, October 2018. Association for Computational Linguistics.

3. Anu Koivunen, Antti Kanner, Maciej Janicki, Auli Harju, Julius Hokkanen, and Eetu Mäkelä. Emotive, evaluative, epistemic: a linguistic analysis of affectivity in news journalism. *Journalism*, February 2021.
4. Eetu Mäkelä, Anu Koivunen, Antti Kanner, Maciej Janicki, Auli Harju, Julius Hokkanen, and Olli Seuri. An approach for agile interdisciplinary digital humanities research — a case study in journalism. In *Twin Talks: Understanding and Facilitating Collaboration in Digital Humanities 2020*. CEUR Workshop Proceedings, October 2020.
5. Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. A Finnish news corpus for named entity recognition. *Lang Resources & Evaluation*, August 2019.
6. Hadley Wickham. Tidy data. *Journal of Statistical Software*, 59(10), August 2014.