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

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 this poster, we present 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.

Our approach rests on three pillars. The first of these is an interface for close reading, but crucially one which is able to highlight to the user all results from automated computational annotation. Beyond pure close reading, through this interface, the user is thus also able to evaluate the quality of computational analysis. Further, the interface supports manual annotation of the material, facilitating correction and teaching of machine-learned approaches.

The second of our pillars is an interface for statistical analysis, where the phenomena of interest can be analyzed en masse. However crucially, this interface is also linked to the close-reading one to further let the users delve into interesting outliers. Through this, they are not only able to derive hypotheses and explanations of the phenomena, but can also identify cases where outliers are more due to errors and omissions in our computational pipeline.

Finally, our third pillar is an agile pipeline to move data between these interfaces and our computational environment. In application, this third pillar is crucial, as it allows us to iteratively experiment with different computational indicators to capture the objects of our interest, with the results quickly making their way to experts for evaluation and explorative analysis. Through this analysis and evaluation, we then equally quickly get back information on not just the technical accuracy of our approach, but also if it captures the question of interest. Further, beside direct training data, we also get suggestions on new phenomena of interest to try to capture.

By maintaining from the start interfaces that allow both computer scientists and social scientists to not only view, but highlight to each other all aspects of the data, we also further a shared understanding between the participants. For example, social scientists are easily able to highlight to the computer scientists new phenomena of interest in the data derived from their close reading, while the computer scientists can easily show what they are currently automatically able to bring forth from the data. Through this, everyone is kept on the same page, misunderstandings are avoided, and the most fruitful avenues for development can be negotiated in a shared space where everyone contributes equally.

Combined with the capability for agile development and experimentation, this provides a versatile template for an iterative and discursive approach to digital humanities research, which moves toward questions of interest both fast, as well as with high capability to truly capture the phenomena from all viewpoints of interest.

In this poster, we present insights into the interaction between close reading and computational methods gained from the work in our current project: *Flows of Power: media as site and agent of politics*. The 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).

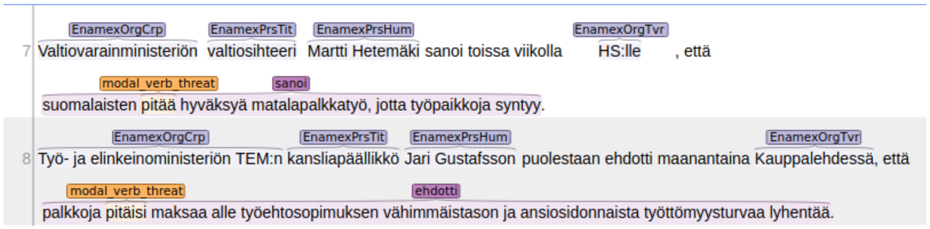
2 Software and data formats

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 [3] for named entity recognition.

Our primary toolbox for statistical analysis is R. This motivates using the 'tidy data' CSV format [4] 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.

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

For visualization, close reading and manual annotation, we decided to employ WebAnno [1].[5] 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.[6] 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. 1). 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.



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

An important advantage is WebAnno's low barrier of entry. It is a Web application, meant to be deployed on a server and used through a Web browser. This kind of usage requires neither technical skills nor any installation on the users' machines. Furthermore, WebAnno contains functionality for managing projects and user accounts. The server application can be also run locally, in form of a JAR file, which is useful for trial and demonstration purposes.

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 the custom annotation layers. Because it is a text format and is well documented, we are 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.

Finally, some automatic annotations are produced by rule-based approaches implemented in Prolog. Thus, another document representation that we utilize

[5] `https://webanno.github.io/webanno/`

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

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. 2.

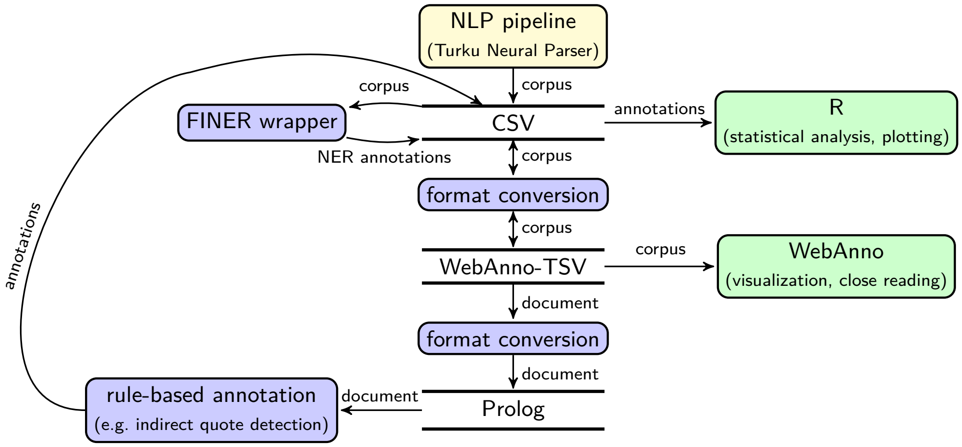


Fig. 2. A dataflow diagram of the back-end.

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 highly subject-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 final step, which is still in progress, is to validate the automatic annotation via close reading of another set of articles. WebAnno’s functionality of highlighting and manual correction of annotations greatly facilitates such work.

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. Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. A Finnish news corpus for named entity recognition. *Lang Resources & Evaluation*, August 2019.
4. Hadley Wickham. Tidy data. *Journal of Statistical Software*, 59(10), August 2014.