Search Your Own Treebank

Alexandr Chernov, Erhard Hinrichs, Marie Hinrichs

Department of Linguistics
University of Tübingen
E-mail: firstname.lastname@uni-tuebingen.de

Abstract

This paper reports on the use of the treebank search and visualization tool TüNDRA for the purposes of inspecting linguistically annotated data that are generated by the web-based annotation tool WebLicht. The motivation for enriching WebLicht by the functionalities offered by TüNDRA is twofold: (i) it allows on-the-fly searches for specific linguistic phenomena at the word and sentence level, (ii) it provides on-the-fly visualizations of such phenomena.

1 Introduction and Motivation

TüNDRA [10] is a web application for hosting, querying, and visualizing treebanks. It currently hosts 63 treebanks, including the Tübingen treebanks for German, the suite of Universal Dependency treebanks [12], the Latin Index Thomisticus treebank, as well as the Perseus treebanks for Latin and Ancient Greek. TüNDRA supports visualization of constituency-based and dependency-based treebanks and uses the TIGERSearch query language [7] for searching and statistical aggregation of treebank data. Recent back-end optimizations of TüNDRA and WebLicht (Web-based Linguistic Chaining Tool) [5] reported in [6] allow WebLicht to create very large treebanks and TüNDRA to host them. An example of this is the automatically annotated dependency treebank of the German Wikipedia, TüBa-D/W, with approximately 36 million sentences and 615 million lexical tokens.

Prior to the research and development reported in this paper, TüNDRA was used chiefly for traditional treebank data, even though its functionality is equally applicable to searching and visualizing linguistic annotations that are generated dynamically by the workflow engine WebLicht. WebLicht is a web application for building and executing natural language processing pipelines. It provides easy access to a wide range of text processing tools (e.g. tokenizers, lemmatizers, part-of-speech taggers, morphology analyzers, parsers, etc.) which can be assembled to form processing chains for incremental annotation of linguistic data. To address the difficulties arising from the fact that each tool has its own input and output formats,

TCF (Text Corpus Format) [4] was developed for use as an internal data exchange format. Annotation tools are wrapped as webservices that receive and return TCF. WebLicht is a distributed system, where the annotation tool webservices are hosted at CLARIN[1] centers and are invoked via HTTP requests. Currently, WebLicht can be used to process 11 different languages with over 100 annotation tools hosted at 9 CLARIN centers.

WebLicht has been generally well accepted and has gained an increasing user base over time, with approximately 700,000 invocations of its webservices in the past calendar year. However, it still lacks the important feature of providing straightforward search functionality on the resulting annotations. Such functionality allows users to better explore their annotation results, enabling them to ask questions about their data, such as:

- How was a particular word form annotated for part of speech?
- How many occurrences of a proper name were successfully annotated as a named entity?
- How many occurrences of a particular syntactic construction were annotated in the data set?

Since TüNDRA provides precisely the type of querying functionality illustrated by the above examples, and WebLicht provides the tools to easily create custom, on-the-fly treebanks, the two applications have been more tightly coupled. The remainder of this paper is structured as follows: Sections 2 and 3 describe the state of visualization and exploration in WebLicht before and after TüNDRA integration, respectively. Section 4 describes enhancements to TüNDRA, including those required for WebLicht integration. Sections 5 and 6 describe related and future work, respectively.

2 WebLicht and TüNDRA Before Integration

In order to better recognize the motivation and impact of the work described here, it is necessary to understand the prior states of both WebLicht and TüNDRA in terms of visualization and search functionality. This section gives some background information about the two applications in isolation.

2.1 WebLicht Before Integration

Prior to the integration of TüNDRA into WebLicht, visualization of annotation results were provided in ways that are appropriate for the individual annotation layers. A table view was used for tokens, lemmas, part-of-speech tags, and morphology annotations. Named entities were highlighted within the text using color-coding to distinguish between different types (person, location, organization, etc).

[1] https://www.clarin.eu/

A graphical view is used for constituency parse trees, and dependency parse trees were displayed using embedded *brat* [14] visualizations. Figure 1 shows examples of these WebLicht visualizations.

Before the integration with TüNDRA, WebLicht provided no direct query functionality for annotation results. Although querying of annotation results produced by WebLicht was in some cases possible, it was cumbersome and did not provide a good user experience. Consider the procedure for performing a search on annotations contained in the table view. First the table needed to be downloaded and opened in external spreadsheet software, followed by use of the generic and rather rudimentary search functionalities of the spreadsheet software.

2.2 TüNDRA Before Integration

Since TüNDRA was specifically designed for processing treebanks, much care was taken to provide visualization and search support for any annotations that a treebank may contain. TüNDRA's flexibility enables it to support nearly any type of treebank. No assumptions are made about what a treebank node represents or the number or type of features it contains. Exactly this flexibility makes it easy for TüNDRA to work with the dynamic data produced by WebLicht, which may or may not have structural information.

TüNDRA uses the query language Tiger [7, 9], which supports querying of both constituent-based and dependency-based treebanks. The Tiger language supports the querying of nodes and of edge labels in syntax graphs. Individual nodes can be identified by hash-tag variables and further specified by Boolean expressions of feature-value pairs. Tiger queries can make reference to the two primitive node relations of precedence (.) and (labelled) dominance (>). The dominance relations can be further specified by particular edge labels. Node identifiers in a query can, inter alia, be used to collect statistics on matches. Although some understanding of the structure and of the features of a treebank are required in order to form search queries, the required information can usually be gained by browsing through the treebank itself or by the stylebook of a treebank, if such off-line documentation is available. See section 3 for examples of using TüNDRA to query and gather statistics on dynamic WebLicht data.

3 Integration of TüNDRA into WebLicht

Using the newly integrated TüNDRA, it is possible to conveniently execute queries and to gather statistics on annotations directly within the WebLicht application. Queries are not restricted to parsed data, but can be executed on all levels of annotation. However, structural queries can only be successful if the text has been parsed, which is not always the case for WebLicht data. This section presents two examples of using TüNDRA to query and gather statistics on WebLicht-generated datasets.

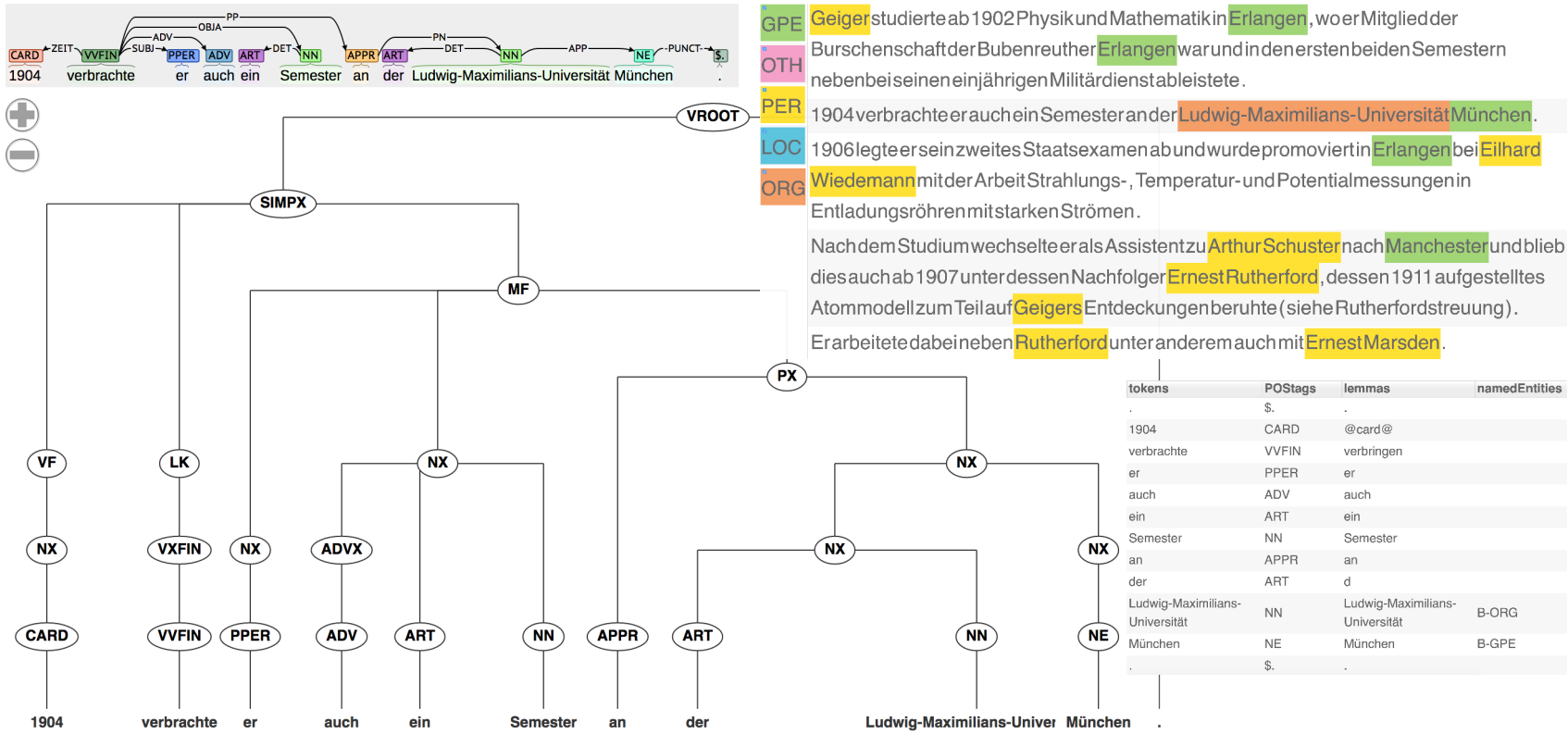

Figure 1: WebLicht Visualizations: Dependency and Constituency Trees, Named Entities, Table View
Including the sentence: *In 1904 he also spent a semester at the Ludwig-Maximilian-University in Munich.*

3.1 Example 1: Searching for Named Entities in a German Corpus

The following example demonstrates gathering statistics about named entities on non-parsed data. The WebLicht data format includes an attribute for named entity annotations for further categorizing them (e.g. person, location, etc). The new TüNDRA implementation makes it possible to search for named entities of a particular type. The query in (1), executed with the "statistics" option, finds the frequency and percentage of geo-political named entities:

(1) `#ne:[_ne="GPE"]`

This query was executed on a text that was automatically annotated with named entities in WebLicht. The text was about the physicist Hans Geiger and the results of the statistical query can be seen in the table in Figure 2.

In addition to the benefit of enabling exploration of WebLicht data, the integration of TüNDRA into WebLicht leads to a unified presentation of data in the applications. Since TüNDRA was designed to process parsed text, it is particularly advantageous to have its constituency-based and dependency-based visualizations in WebLicht. Figure 2 shows parse tree visualizations as they appear in both WebLicht and TüNDRA.

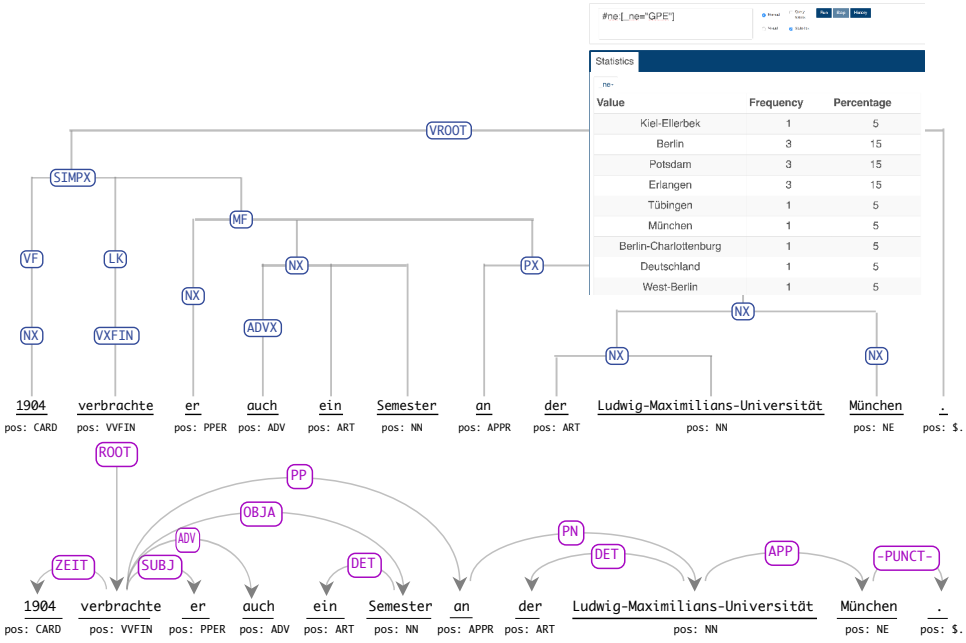

Figure 2: Sample Trees and Statistics View for Example 1

3.2 Example 2: Searching for Verb-Preposition Pairs in an English Corpus

Since the WebLicht suite of services is not limited German, but also includes tools for English, it is also possible to annotate and query English corpus data. The query used in section 3.1 is simple in the sense that it only searches for a single class of lexical tokens. However, TüNDRA supports the full expressiveness of the Tiger query language, thus allowing also queries that involve multiple constituents and/or lexical tokens. For example, lexicographers and lexical semanticists may be interested in the set of prepositions that co-occur with a particular verb in a given corpus. The query in (2) exemplifies such a query for the English verb *agree* in a constituency-based treebank.

```
(2)  #vp:[cat="VP"] > [lemma="agree"]
     & #vp > [cat="PP"] > #p:[pos="IN"]
```

This query searches for VP nodes (`#vp`) which dominate a lexical node with lemma *agree* and a PP node, which in turn dominates a lexical node with part-of-speech `IN`, the label used for prepositions in the Penn treebank tagset [8]. Since this lexical node is identified by the hashtag variable `#p`, statistics on the word forms for this lexical node can be gathered.

The same type of query can be created for execution on a dependency treebank. The following query finds lexical nodes with lemma *agree* and with an edge, labelled with relation name `VMOD` and pointing to a lexical node with part-of-speech label `IN`.

```
(3)  [lemma="agree"] > VMOD #p: [pos="IN"]
```

Figure 3 shows a sample sentence found by the constituency and dependency queries in the Leipzig University Corpora Collection [3], which was annotated with lemmas, part-of-speech tags, constituent and dependency parsing in WebLicht. In both the constituency and dependency annotation in Figure 3 the portion of the constituency and dependency annotation that matches the Tiger query is highlighted in the TüNDRA visualization for ease of reference. In addition, Figure 3 shows the list of prepositions, sorted by absolute frequency, that co-occur with the verb *agree* in the corpus at hand.

4 Enhancements to TüNDRA

In order to complete the integration of TüNDRA into WebLicht, it was necessary to (i) accommodate the WebLicht data format and (ii) refactor the code into front-end and back-end components.

Before WebLicht data can be used in TüNDRA, it must first be converted into the TüNDRA internal format. This is a straightforward process for most annotations at the token level, with the exception of named entities, which can span more than one token. They are handled by adding a special "_ne" attribute. At the structural level, the case where parse annotations are not present in the data must be

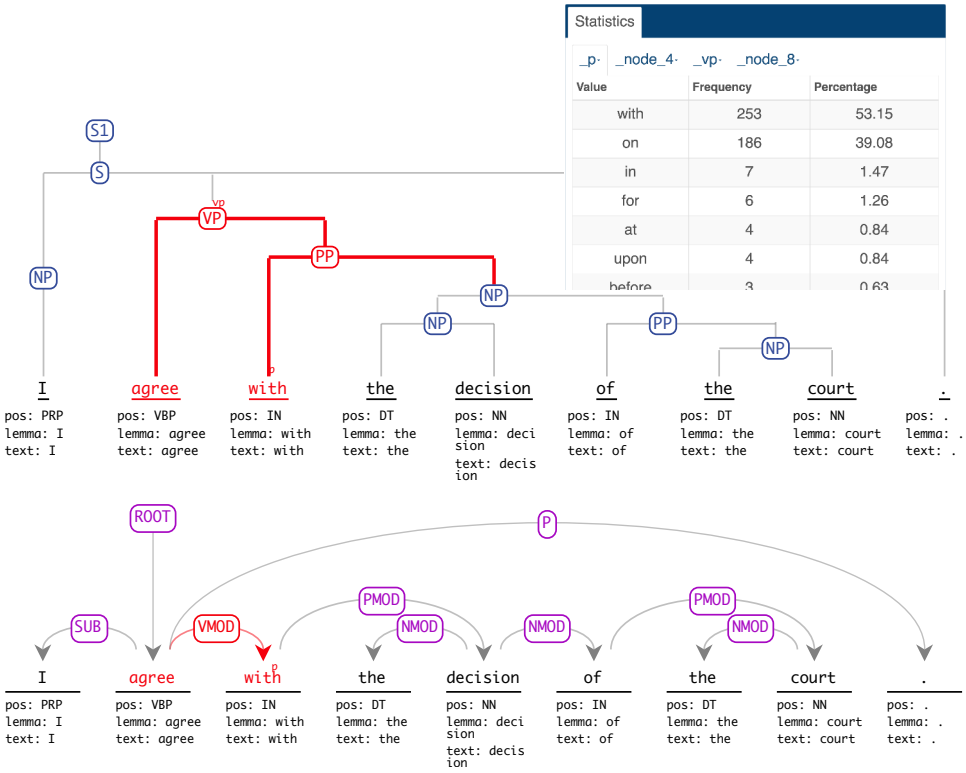

Figure 3: Sample Trees and Statistics View for Example 2

handled. This is done by creating "fake" tree structures with all tokens attached directly to the root, allowing them to be processed like all other trees in TüNDRA. In addition, a table view, similar to that which was previously available in WebLicht, but not in TüNDRA, has been incorporated into TüNDRA.

It was necessary to refactor the TüNDRA code into well-defined front- and back-ends. The front-end is used by both applications. This is done in WebLicht by simply replacing the prior visualization component with TüNDRA's front-end. The front-end in turn communicates with the back-end which does any necessary data conversion and performs queries. This clean division of labor into the user interface (front-end) and the query-processing engine (back-end) made it possible for both applications to share the same visualization and search component.

5 Related Work

There are many well-known treebank search and visualization tools, such as INESS [11], PML-TQ [13], GrETEL [1], and ICARUS [2]. Although some of them provide upload or import options for processing personal treebanks (PML-TQ,

ICARUS), to our knowledge none of them are tightly integrated into a workflow engine that allows on-the-fly annotation using custom-built pipelines. INESS has rich support for search and visualization of hosted treebanks, including comparison of parallel corpora, but only authorized users can run annotation pipelines. GrETEL is a treebank search engine which is very easy to use due to its novel way of guiding the query building process, but only supports hosted treebanks.

6 Conclusion and Future Work

In this paper we have shown how the treebank visualization and search application TüNDRA has been integrated into the annotation workflow engine WebLicht, making it possible to apply TüNDRA's full query/visualization/statistics capabilities to on-the-fly treebanks created in WebLicht. It was necessary to refactor large parts of the TüNDRA code to make its front-end available in WebLicht. At the time of writing, the new versions of both applications are in beta phase.

The main focus of future work in TüNDRA are in the areas of query building and statistics views. Query building needs to be simplified for users who are unfamiliar with the query language. This can be done by offering graphical guidance and limiting elements of the query to valid values where possible. It is also planned to provide more detailed statistics views, including more visualization options and allowing more in-depth exploration of the statistics. A version of TüNDRA that can be run and administered locally is also planned, enabling the local use of treebanks that, for example, cannot be hosted by the public version of TüNDRA for legal reasons. In the near future, the public treebanks hosted by TüNDRA will be made available without the need for logging in.

Future work on WebLicht includes providing a batch mode for more convenient processing of very large texts. This goes beyond what WaaS can already do (executing chains from the command line or programming code) by splitting up very large texts into smaller chunks for processing if necessary, invoking the tool chain in parallel on the smaller chunks, piecing it all back together, and storing the finished result for later download. Users will be able to monitor the progress of their jobs and will be notified when it is finished.

References

[1] Liesbeth Augustinus, Vincent Vandeghinste, and Frank Van Eynde (2012). Example-based Treebank Querying. In: *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC-2012)*. Istanbul, Turkey. European Language Resources Association (ELRA).

[2] Markus Gärtner, Gregor Thiele, Wolfgang Seeker, Anders Björkelund and Jonas Kuhn (2013). ICARUS – An Extensible Graphical Search Tool for Dependency Treebanks. In: *Proceedings of the 51st Annual Meeting of the*

Association for Computational Linguistics: System Demonstrations. Sofia, Bulgaria.

[3] Dirk Goldhahn, Thomas Eckart und Uwe Quasthoff (2012). Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In: *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC-2012)*. Istanbul, Turkey. European Language Resources Association (ELRA).

[4] Ulrich Heid, Helmut Schmid, Kerstin Eckart, and Erhard 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 (LREC 2010)*. Valletta, Malta. European Language Resources Association (ELRA).

[5] Erhard Hinrichs, Marie Hinrichs, and Thomas Zastrow (2010). WebLicht: Web-Based LRT Services for German. In: *Proceedings of the Systems Demonstrations at the 48th Annual Meeting of the Association for Computational Linguistics (ACL-2010)*. pp. 25-29. Uppsala, Schweden.

[6] Daniël de Kok, Dörte de Kok, and Marie Hinrichs (2014). Build your own treebank. In: *Proceedings of the CLARIN Annual Conference*. Soesterberg, The Netherlands

[7] Wolfgang Lezius (2002). TIGERSearch - Ein Suchwerkzeug für Baumbanken. In: *Proceedings der 6. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2002)*. Saarbrücken.

[8] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz (1993). Building a Large Annotated Corpus of English: The Penn Treebank *Computational Linguistics*, 19.2, pp. 313–330.

[9] Scott Martens (2012). TüNDRA: TIGERSearch-style treebank querying as an XQuery-based web service. In: *Proceedings of the joint CLARIN-D/DARIAH Workshop "Service-oriented Architectures (SOAs) for the Humanities: Solutions and Impacts" (DH 2012)*. Hamburg, pp. 41-50.

[10] Scott Martens (2013). TüNDRA: A Web Application for Treebank Search and Visualization. In: *Proceedings of The Twelfth Workshop on Treebanks and Linguistic Theories (TLT12)*. Sofia, pp. 133—144. (URL: http://bultreebank.org/TLT12/TLT12Proceedings.pdf)

[11] Paul Meurer (2012). INESS-Search: A Search System for LFG (and Other) Treebanks. In: Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG2012 Conference*. CSLI Publications.

[12] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty and Daniel Zeman (2016). Universal Dependencies v1: A Multilingual Treebank Collection. In: *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA).

[13] Petr Pajas and Jan Štěpánek (2009). System for Querying Syntactically Annotated Corpora. In: *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 33–36, Suntec, Singapore. Association for Computational Linguistics.

[14] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou and Jun'ichi Tsujii (2012). brat: a Web-based Tool for NLP-Assisted Text Annotation. In: *Proceedings of the Demonstrations Session at EACL 2012*.