Treebanks for the Ordinary Working Grammarian

Joel Priestley[1], Anders Nøklestad[1], Kristin Hagen[1], Anu Laanemets[2] and Dag Trygve Truslew Haug[1,2]

[1]Humit - Centre for Digital Development, University of Oslo, Norway
[2]Department of Linguistics and Scandinavian Studies, University of Oslo, Norway

Abstract

In this paper we present how three treebanks of Norwegian have been incorporated in the Glossa search interface, allowing users without specialized training to formulate queries based on syntactic information. One of the treebanks contains written material (mostly newspaper text, but also blogs, magazines and other genres) and the two other treebanks are based on transcriptions of spoken dialects. The user interface is simple and only allows access to selected features of the annotation. We show through two case studies how it can nevertheless be useful for the large group linguists who do not have the time or inclination to learn a full treebank query language. We argue that our tool fills an important gap and can help bring treebank data to new users.

Keywords

corpora, query interfaces, syntax

1. Introduction

By now, text corpora are a standard tool in linguistics, found across most subdisciplines from historical linguistics to theoretical syntax. Typically, the corpora that are used consist of raw text with rich metadata (with features such as genre, dialect, date, author gender and age and much more) and some linguistic annotation such as part of speech tags. Numerous tools and web pages are available that make the exploration of such data easy without specialized training, such as Sketch Engine, the Coca web interface, and – especially for Norwegian corpora – the Glossa web interface.[1]

Until recently, more complex linguistic annotation, such as syntax, was only found in a few specialized resources such as the Penn Treebank [8]. But over the last decade, the Universal Dependencies (UD) framework [14, 2] for annotating dependency syntax has spurred the creation of many more treebanks. Currently more than 200 treebanks for more than 150 languages are

CHR 2024: Computational Humanities Research Conference, December 4–6, 2024, Aarhus, Denmark
joeljp@uio.no (J. Priestley); noklesta@uio.no (A. Nøklestad); kristiha@uio.no (K. Hagen); anu.laanemets@iln.uio.no (A. Laanemets); daghaug@uio.no (D. T. T. Haug)
0000-0001-5275-8073 (D. T. T. Haug)

[1]The Glossa web interface is hosted at at the University of Oslo, along with a number of corpora, at the following URL https://tekstlab.uio.no/glossa3/ Access is provided by Clarin (alternatively Feide for Norwegian organisations). Non-Clarin affiliated users can apply for access by sending an email to `tekstlabpost@iln.uio.no`. Although the majority of corpora in Glossa are currently Norwegian UTF-8 encoded corpora, Glossa can easily handle any language in any text encoding. The Glossa source code can be cloned from GitHub at https://github.com/textlab/fglossa.

available on the Universal Dependencies webpage.[2] But there are few if any tools available that make treebanks accessible to ordinary humanities scholars without special training.

Partly this is due to the inherent complexity of syntactic annotation, which is not reducible to attributes of words, but involve labelled relations between words that give rise to a nested structure. Most tools that let users explore syntactically annotated corpora are therefore based on a tree (or graph) description language such as INESS [9], Grew [5] or Semgrex [1]. These let the user specify arbitrarily complex constraints on the syntactic structure, but the learning curve is often steep.

In this paper, we describe a different approach, where we offer easy access to some important aspects of syntactic annotation within Glossa, a corpus tool focused on user-friendliness, where all queries can be done with a combination of dropdown menus and a google-like text search box. We describe how three treebanks of Norwegian, one with written language texts and two with spoken language texts, have been imported in Glossa and show with a number of case studies that interesting queries can be formulated even in this simpler setting. While acknowledging that in-depth studies of treebank data will require more advanced tools, we argue that easy-access tools like ours allow a broad audience of linguists and other humanities scholars to make use of data that otherwise would be out of reach.

2. The data

At present there are three treebanks imported into Glossa: The Norwegian Dependency Treebank (NDT, [13]), The LIA Treebank (LIA, [11]) and The Nordic Dialect Corpus Treebank (NDC, [6]). Norwegian Dependency Treebank (NDT) has two parts, one for text written in the Norwegian standard ”Nynorsk” and one for ”Bokmål”.[3] LIA and NDC are treebanks with transcriptions of Norwegian spoken dialects, LIA with transcriptions in Nynorsk, NDC with transcriptions in Bokmål.

NDT, LIA and NDC are all dependency treebanks comprising words annotated with morphological features, syntactic functions, and hierarchical structures. The treebanks are available for download in CoNLL format. The annotations were made with different automatic tools, but every annotation was subsequently proofread and corrected by one or two linguists, see the references for details of each treebank.

NDT Nynorsk and NDT Bokmål have approximately 300.000 tokens each, collected from newspapers, magazines, and blogs. For the most part, the annotations follow the analyses in The Norwegian Reference Grammar [4], but detailed annotation guidelines were also developed to document the dependency grammar analyses.[4] The Norwegian Dependency Treebank was developed by Språkbanken at the National Library.

The LIA Treebank includes 7,536 speech segments and 77,701 tokens from transcriptions in Nynorsk from the speech corpus LIA Norwegian - Corpus of historical dialect recordings. The recordings in the treebank took place between 1958 and 1981, and the 41 speakers come from 21 places in different dialect areas of Norway.

[2] https://universaldependencies.org/

[3] ”Nynorsk” and ”Bokmål” are two different written standards for Norwegian.

[4] https://www.nb.no/sbfil/dok/20140314_guidelines_ndt_english.pdf

The NDC Treebank contains 4,637 speech segments and 66,042 tokens from the Bokmål transcriptions in the Norwegian part of the Nordic Dialect Corpus. The recordings took place between 2007 and 2010, and the 43 speakers come from 17 places in the same dialect areas as the speakers in the LIA Treebank.

Both the LIA and the NDC Treebank have been transcribed in two ways, one (quasi) phonetically and one orthographically. In the Glossa interface you can search both transcriptions. On the results page you can also listen to and watch (there are video recordings in NDC) the original recordings for the search results.

Since spoken language contains speech features like pauses, unfinished/incomplete words and disfluencies such as repairs and deletions, we had to adapt and add to the NDT guidelines to cover transcription and annotation of the spoken treebanks. For example, the transcribed texts are divided into speech segments. A speech segment is our spoken language approximation of a sentence. Speech segments can lack otherwise required syntactic features like verbs and subjects, or they can contain only adverbials or interjections. Pauses are transcribed simply as # and ## and incomplete words are written as they are spoken, demarcated with a hyphen ("-") and given the morphological label "ufullst" (short for incomplete). Repairs and deletions get their own syntactic labels: REP and SLETT (delete). For a more detailed description, see [11, 6].

3. Syntax in Glossa

Glossa is a user-friendly and functional search interface developed and upgraded at the University of Oslo over the past twenty years [10]. Glossa is used for more than 40 written, multilingual and speech corpora. The easiest option for search is to write one or more words in a Google-like search box and filter the results by a metadata menu on the left. The results are given as concordances, and for speech corpora the results are linked to audio and video. There is also an easy-to-use extended search box with clickable boxes and menus to be used for finding e.g. lemmas or part of speech and other morphological information, see Figure 1, as well as an option to access the underlying query language (CQP, see below). Search results can be exported in Excel or CSV/TSV format. In the following we describe how Glossa was adapted to be able to handle treebanks with syntactic information.

The query engine used in Glossa is the Corpus Query Processor (CQP) from the IMS Open Corpus Workbench (CWB; [3]). Although the CWB has some limited support for (non-recursive) structural attributes, it is mainly geared towards searching in token-level annotation. This makes dependency grammar a better fit for Glossa than phrase structure grammars or other types of grammar based on hierarchical structures.

For the treebanks in Glossa, information about dependents and heads is fetched from files in CoNLL format and stored in positional CWB attributes, i.e., attributes associated with individual tokens. The CWB format uses XML tags for structural attributes, such as sentences, speech segments, etc. These tags encompass a one-word-per-line, tab-separated representation of texts, where each column holds a specified annotation. The additional annotations from the CoNLL are simply appended as successive columns: Function, Index (1-based, as 0 dependency implies root status) and Dependency.

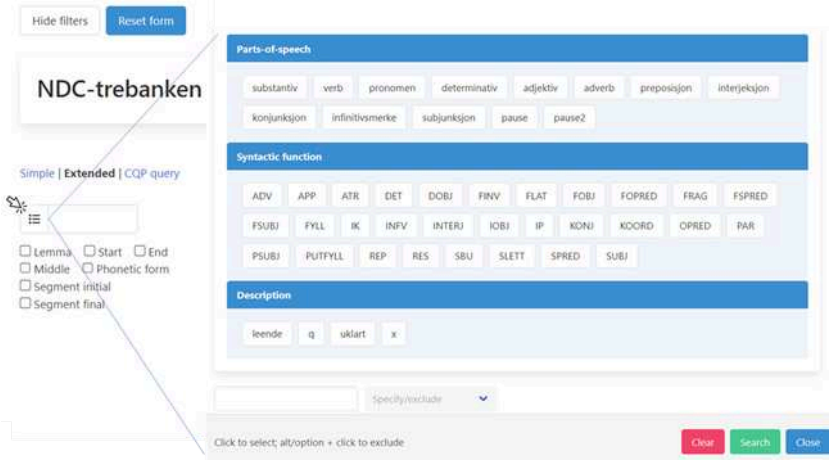

Figure 1: The Glossa Extended search box with the search menu for part of speech and syntactic categories

While these extra annotations are taken directly from the CoNLL files, a fourth attribute is derived, namely syntactic level, which is assigned one of three clause types: Main, Dependent or Infinitive. To achieve this, a sentence is scanned to identify verbs in non-root positions, i.e., verbs with a non-zero Dependency value. Such verbs are then either tagged as Dependent or Infinitive, according to their morphosyntactic features. A simple parse of the sentence is then performed to create a tree structure. Starting at the identified node in the parse tree, all nodes in its subtree can then be given the same syntactic level tag. All other nodes receive the Main tag.

There are two alternatives for accessing the new layers of annotation in Glossa. Queries can be formulated directly in the CQP query field. This requires some basic knowledge of regular expressions as well as CQP specific features. A simple search for the token ”mellom” within a dependent clause, for example, would be as follows: `[word="mellom" %c & niv="led"]`.

A more intuitive option is to use the extended menu, which lists all available attributes, such as part of speech, their relevant morphosyntactic features, as well as syntactic functions and level, as shown in Figure 1. A benefit of using this method is that the risk of selecting mutually exclusive attributes is removed.

Trees are rendered with SVG (Scalable Vector Graphics). Glossa packs CQP output into a JSON object which is passed to a React component. The component first groups dependent nodes according to their proximity. Adjacent nodes will be assigned the lowest arching edges. Height is increased with distance, avoiding edges crossing and enhancing readability. Once this is done, the component plots the nodes and joining edges using the SVG path and text elements. For the highlighting effect, a mouseover/mouseout event listener is added to each node. A function provided to the listener will, when triggered, traverse the chain of dependencies, up to the root, adding or removing CSS styling as appropriate. The resulting SVG object is then returned to Glossa and rendered as in Figure 2.

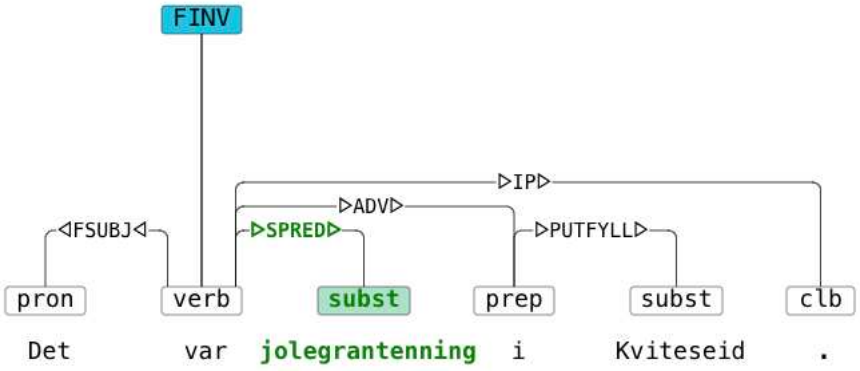

Figure 2: Syntactic tree in Glossa

4. Case studies

In this section we present two case studies to demonstrate the possibilities and benefits of these new query functions.

4.1. Case 1 – subject complement construction

Typical examples in a grammar book on subject complement constructions [7] are as in (1).

(1) a. De er raske.
they are fast.PL
b. Ho er feminist.
she is feminist

In a subject complement construction, the typical complement is either an adjective, as *raske* ('fast') in example (1-a), or a noun, as *feminist* in example (1-b), and is linked to the subject by a copula/linking verb (most typically *være* ('be')). As is apparent, the construction involves large word classes (adjectives and nouns), which may have many different functions in a sentence and the verb *be*, which is both a high frequency verb in a text and can have different functions in a sentence. Thus, if we are interested in studying subject complement constructions in a corpus with only part of speech and morphological annotation, we will most probably get a lot of examples with no relevance for the study as the possibilities to narrow down the query in an appropriate way are sparse. The best we could do would be to search for the lemma *være* ('be') followed by an empty slot for any kind of word (e.g., a modifier or an adverb) followed by an adjective. In Glossa, this query can be expressed with Google-like search boxes and dropdown menus. By clicking on the CQP query link, we get the translation to CQP shown in (2).

(2) `[lemma="være" %c] []{,1} [pos="adj"]`

Clearly, the Glossa interface makes it much easier to express the query. Nevertheless, the results are not very precise. This query results in 4046 matches in the NDT treebank (Bokmål, 311,277 tokens). A quick look at the examples reveals that many of them turn out not to be

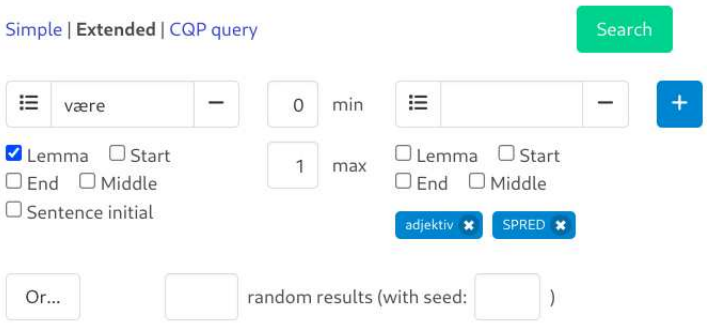

Figure 3: Glossa query with lemma and morphology

relevant. More precisely, among the first 50 matches 24 were not relevant. If we replace the adjectival complement (`pos="adj"`) in the CQP query exemplified in (2) with nominal complement (`pos="noun"`) and repeat the search, we get 2443 matches. Here again, 25 matches out of the first 50 turn out not to be relevant. As the search results show, roughly every second match is actually not relevant for the research purpose. Such a high percentage of irrelevant examples necessitates a lot of manual sorting afterwards in order to avoid misleading conclusions. In cases like this, the possibility to search on syntactic functions will reduce the number of irrelevant examples.

We can take the query string in (2) and add the specification of syntactic function SPRED (subject complement, in Norwegian subjektspredikativ), by simply clicking the function box. This yields the query shown in Figure 3, and if we click the CQP query link, the query specified in the GUI gets translated to CQP as in (3).

(3) `[lemma="være" %c] []{,1} [(pos="adj") & (fun="SPRED")]`

Again, because the syntactic representation is simplified to attributes on words, it can easily be accommodated within the Glossa graphical interface.

The new, specified search results in 2644 matches, which means a reduction by 1402 matches, i.e., 35%. The same specification of syntactic function can be added to the search with nominal complements. Here again we see a significant reduction of matches (from 2443 to 1237, i.e., a reduction by approx. 50%).

The possibility to specify the syntactic function also provides several other search options. One could just search on the syntactic function with no other specifications (CQP query: `[fun="SPRED"]`). When using this query, we get a concordance list of all subject complements in the NDT treebank. Then we can combine the list of concordances with other functions in Glossa like the calculation of frequencies based on e.g. part of speech. By doing so we can find out that many other words can be heads in phrases that constitute the subject complements, such as prepositions, pronouns, determiners, adverbs, and even infinitive and nominal clauses. We can also find out which other verbs, besides the most typical *være* ('be') and *bli* ('become'), can link a subject complement. The following search query – `[pos="verb"] [fun="SPRED"]` – reveals

that nearly one hundred different verb lexemes can link a subject complement. This shows the usefulness of even a simplified syntactic representation reduced to attributes of words, which moreover has the advantage of being easily queried in a graphical interface, as shown in Figure 3, without the need for a specialised query language.

4.2. Case 2 – conditional clauses with inversion

In the second case study, we will demonstrate the benefits of yet another new query function, namely the specification of syntactic level as either main, dependent, or infinitive clause. We will illustrate this function by looking at a special conditional clause construction in Norwegian. In Norwegian, there are two basic word orders, one typically used in main clauses, and one typically used in subordinate clauses. In subordinate clauses, the subject and the adverbial precede the finite verb like in the conditional clause illustrated in (4).

(4) Hvis han ikke har rett, er saken avgjort.
if he not has right is case-DEF settled
'If he is not right, the case is settled.'

In Norwegian, there is another possibility to express a conditional clause without the explicit *hvis* ('if') and with an inverted word order as illustrated in (5).

(5) Har han ikke rett, er saken avgjort.
has he not right is case-DEF settled
'If he is not right, the case is settled.'

This word order is similar to the word order in main clauses, especially so with yes/no interrogatives as illustrated in (6):

(6) Har han ikke rett?
has he not right
'Isn't he right?'

If you are interested in studying this special type of conditional clauses (without the initial *hvis* and with the finite verb first) in a corpus with only morphosyntactic annotations, the possibilities to narrow down the search query are limited and you will most probably end up with many irrelevant examples, as e.g., yes/no interrogatives, which also have a verb-initial word order. In Glossa, we can formulate an extended search by selecting the category 'verb' among the part of speech tags and then tick off for 'sentence initial' position. A CQP query for this search looks like in (7):

(7)
```
<s>[pos="verb"]
```

A search like this results in 778 matches. As predicted, the results include many irrelevant examples as yes/no interrogatives, but also imperatives and incomplete clauses with omitted subjects which are often used in newspaper headings. Because many corpora contain a large quantity of newspaper text, the latter category is not insignificant. As described before, the new

annotation layers implemented in Glossa provide the possibility to search for a construction in either main clauses, subordinate clauses or infinitive clauses. We can restrict the query to subordinate clauses with the attribute `niv="led"` added to the search string in (7), as illustrated in (8), and repeat the search.

(8) `<s>[(pos="verb") & (niv="led")]`

This search results in 78 matches. A closer look at the examples confirms that the vast majority of them are relevant for our purpose, that is, they are conditional clauses with inversion. This means that the new query function significantly reduces the number of irrelevant examples, leaving us with only around 10% of the initial search result. Or to put it another way, without the new function we got a results list where only about every tenth match was relevant to our purpose, while the rest would have had to been sorted out manually.

5. Outlook

We have shown how a suitably simplified syntactic representation can easily be queried in the Glossa search interface tool, while still remaining useful for many queries. While there are real limits to what can be done (e.g. one cannot simultaneously constrain the features of a dependent and its head) compared to what is possible in full tree-based query formalism, our tool is also much easier to learn. Even compared to a tool like Treebank.info ([12]), which like the current project is built on CWB, and which also allows the use of menus and other graphical elements to specify a query, our system still seems considerably simpler to learn, with a correspondingly narrower scope. This makes complicated syntactic data accessible to ordinary linguists without specialised training and thereby opens up for more widespread use of a treebank data in linguistics. The recent surge in creation of treebanks has not seen a corresponding increase in the use of the data, partly we think for accessibility reasons. Recruiting users through the simple Glossa interface may in time even increase the interest in full-fledged query languages.

There are also many other types of annotation that are often created with an eye towards NLP applications, but that could be useful for general linguists as well. For the Norwegian Dependency Treebank, this includes named entity recognition, animacy and coreference annotations. Each of these annotations introduce their own complexities. Coreference works across sentences for example, while animacy can be assigned at token level, but also phrase level, and be nested, so that a token can participate in animacy on multiple levels. In future work, we plan to address this complexity and integrate these annotation layers into the Glossa query interface to make them similarly accessible to a wider community.

References

[1] J. Bauer, C. Kiddon, E. Yeh, A. Shan, and C. D. Manning. "Semgrex and Ssurgeon, Searching and Manipulating Dependency Graphs". In: *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*. Washington, D.C., 2023.

[2] M.-C. De Marneffe, C. D. Manning, J. Nivre, and D. Zeman. "Universal dependencies". In: *Computational linguistics* 47.2 (2021), pp. 255–308.

[3] S. Evert and A. Hardie. "Twenty-first century Corpus Workbench: Updating a query architecture for the new millennium". In: *Proceedings of the Corpus Linguistics 2011 conference*. Birmingham, 2011.

[4] J. T. Faarlund, S. Lie, and K. I. Vannebo. *Norsk referansegrammatikk*. Oslo: Universitetsforlaget, 1997.

[5] G. Guibon, M. Courtin, K. Gerdes, and B. Guillaume. "When collaborative treebank curation meets graph grammars". In: *LREC 2020-12th Language Resources and Evaluation Conference*. Marseille, 2020.

[6] A. Kåsen, K. Hagen, A. Nøklestad, J. Priestly, P. E. Solberg, and D. T. T. Haug. "The Norwegian Dialect Corpus Treebank". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France, 2022.

[7] S. Lie. *Innføring i norsk syntaks*. Oslo: Universitetsforlaget, 2003.

[8] M. Marcus, B. Santorini, and M. A. Marcinkiewicz. "Building a large annotated corpus of English: The Penn Treebank". In: *Computational linguistics* 19.2 (1993), pp. 313–330.

[9] P. Meurer, M. Butt, and T. H. King. "INESS-Search: A search system for LFG (and other) treebanks". In: *Proceedings of the LFG'12 Conference, LFG Online Proceedings*. 2012.

[10] A. Nøklestad, K. Hagen, J. Bondi Johannessen, M. Kosek, and J. Priestley. "A modernised version of the Glossa corpus search system". In: *Proceedings of the 21st Nordic Conference on Computational Linguistics*. Gothenburg, 2017.

[11] L. Øvrelid, A. Kåsen, K. Hagen, A. Nøklestad, P. E. Solberg, and J. B. Johannessen. "The LIA Treebank of Spoken Norwegian Dialects". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan, 2018.

[12] T. Proisl and P. Uhrig. "Efficient Dependency Graph Matching with the IMS Open Corpus Workbench". In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey, 2012.

[13] P. E. Solberg, A. Skjærholt, L. Øvrelid, K. Hagen, and J. B. Johannessen. "The Norwegian Dependency Treebank". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland, 2014.

[14] D. Zeman et al. *Universal Dependencies 2.14*. 2024. URL: http://hdl.handle.net/11234/1-5502.

.