An Open Infrastructure for Advanced Treebanking

Victoria Rosén[*§], Koenraad De Smedt[*], Paul Meurer[§], Helge Dyvik[*§]

University of Bergen[*] and Uni Research[§]
Bergen, Norway
victoria@uib.no, desmedt@uib.no, paul.meurer@uni.no, dyvik@uib.no

Abstract

Increases in the number and size of treebanks, and the complexity of their annotation, present challenges to their exploration by the research community. Adhering to different formalisms, lacking clear standards, and requiring specialized search and visualization and other services, treebanks have not been widely accessible to a broad audience and have remained underexploited. The INESS project is providing the first infrastructure integrating treebank annotation, analysis and distribution, bringing together treebanks for many different languages, spanning different annotation schemes and including parallel treebanks. The infrastructure offers a uniform interface, interactive visualizations, leading edge search capabilities and high performance computing.

1. Introduction

Treebanks have without any doubt become one of the most powerful kinds of language resources. Parsers with probabilistic components trained on treebanks are now regarded as indispensable for wide coverage analysis and are therefore a prerequisite for realistic applications such as high quality syntax-based machine translation. Moreover, treebanks have a potentially wide user group including also general linguists and language scholars, but these groups still face obstacles in accessing the knowledge embedded in the resources.

The Penn Treebank (Marcus et al., 1993) has been influential as a standard resource and benchmark during the past couple of decades. Treebanks have been developed for a large number of languages, they have become larger in size, and their linguistic annotation is becoming richer, although treebanks with highly detailed syntactic and semantic analyses are still scarce. The field has some *de facto* standards for simple treebank formats but lacks comprehensive technical and organizational solutions for handling the variety of different formalisms, annotation standards and encodings.

Furthermore, the produced treebanking resources and tools are scattered on many sites, each with their own access policies and formats, and some lack curation and archiving policies. By way of example, we mention the German Tiger corpus and the TigerSearch tool which are potentially very useful but are no longer maintained by the creators.

An open infrastructure for the curation and dissemination of treebanks is therefore a timely goal. By 'infrastructure' we mean a persistent, integrated and managed set of services combining data and tools. By 'open' we mean that the system is not limited to a narrow set of data or users, that any researcher in principle can deposit, access and process data. In establishing such open infrastructures, we are moving from passive repositories towards online eScience laboratories which are easy to access and can address a variety of user needs.

Some usage scenarios of very large treebanks have been explored in the Dutch LASSY project, which has produced huge parsed corpora (van Noord, 2009). One such scenario relates to an investigation of conditions on extraposition, a construction where a constituent is discontinuous (cf. the English example *The question [is raised] why the government does not fund more research*). Since it is difficult to find hard rules governing the conditions under which extraposition can apply, an empirical investigation may be in order. However, such an investigation will hardly be possible in plain text corpora, not even in corpora tagged with parts of speech. Crucially, only a syntactically analyzed corpus provides the required level of detail that allows a systematic search with reliable results, as convincingly demonstrated by van Noord (2009). Enabling linguists and other users to address such questions in a user-friendly way across treebanks with different annotations and for different languages, but using a single access point, is a worthwhile goal.

Building a high quality treebank always requires a big investment, as human contributions in the form of linguistic insight and manual quality control are indispensable in addition to automatized procedures. It is therefore important to get a high return on investment by securing the usability of the finished treebank. More and more attention is being paid to archiving and disseminating treebanks, with appropriate documentation and licenses. Some approaches are illustrated by the Prague Dependency Treebank (Hajič, 1998; Böhmová et al., 2003) and the Icelandic Parsed Historical Corpus (IcePaHC) (Wallenberg et al., 2011). The former is distributed by the LDC and browsable on the web, has a bespoke license, and is searchable with an downloadable application (TrEd 2.0) as well as through a client-server application (Netgraph). IcePaHC is archived with versioning, has direct open download links, is released under a LGPL licence and is searchable with the downloadable CorpusSearch 2. Neither is fully accessible through web-based services from a browser.

2. INESS

The INESS project[1], running from 2010 to 2015, is probably the first large scale project aimed at building an eScience infrastructure for the exploration of syntax and semantics based on treebanking, with a wide range of resources and

[1] Infrastructure for the Exploration of Syntax and Semantics, http://iness.uib.no

services. This infrastructure has been operational on an experimental basis since 2010 and has been steadily expanded and adapted. It is not only the project's aim to make it easy for the R&D community to find, filter and download treebanks, but also to let the community actively participate in uploading and annotating treebanks. Furthermore, one of our goals is to provide a more uniform treatment of treebanks so that they can be linked and explored in similar ways.

The most general characteristic of the INESS infrastructure is that its services are fully accessible through any modern web browser, without the need to download and install any other software on the user's platform. The server middleware was written in Common Lisp on top of an open source web server in the same language.[2] The use of the same high level programming language throughout the whole system has resulted in a highly flexible system in which all annotation and analysis services are seamlessly integrated. The system is easy to modify at all levels, which promotes a fast evolution in response to user needs. Visualizations are based on Scalable Vector Graphics (SVG), which is supported by modern web browsers. The remainder of this article will present the status of the INESS infrastructure.

3. Selection of treebanks

While the INESS project is partly devoted to developing a large treebank for Norwegian, the infrastructure is open to hosting any other treebanks which may be useful in research. Currently the INESS middleware can handle LFG, constituency and dependency treebanks in various formats. INESS invites treebanking projects to deposit their treebanks in the infrastructure in order to make them accessible, and it currently provides access to 53 treebanks, ranging from small test suites to full size treebanks. This steadily growing number has made it necessary to provide a search interface at the metadata level. The user can make a choice of treebanks by selecting values for the following criteria:

- Language: All · Norwegian Bokmål (11) · German (6) · Georgian (5) · Hungarian (4) · Latin (4) · Church Slavic (3) · Ancient Greek (to 1453) (3) · Icelandic (2) · Northern Sami (2) · Wolof (2) · Classical Armenian (2) · Abkhazian (1) · Danish (1) · Estonian (1) · Gothic (1) · Norwegian Nynorsk (1) · Swedish (1) · Tigrinya (1) · Turkish (1) · Urdu (1)
- Collections: All · GeoGram (3) · HunGram (4) · IcePaHC (1) · NorGram (8) · PROIEL (13) · Sofie (8) · Test (5) · TiGer (3) · XPar (3)
- Annotation types: All · lfg (30) · dependency-proiel (13) · constituency (8) · dependency-cg (2)

For the languages, ISO-639-3 codes are internally used.

The annotation types currently distinguish between the following: *lfg* (Lexical Functional Grammar); *dependency-proiel*, a dependency annotation used in the PROIEL project,[3] based on dependency grammar enriched with secondary dependencies reminiscent of the structure sharing mechanism in LFG; *constituency*, which provides simple phrase structure constituency; and *dependency-cg*, based on Constraint Grammar.

Collections are loosely defined groups of treebanks based on similar texts or on similar grammars used in the analysis. For instance, *GeoGram* is a collection of materials parsed with the same Georgian grammar while *HunGram* is a collection parsed with the same Hungarian grammar. *Sofie* is a collection based on text from the novel *Sofies verden* [Sophie's World] (Gaarder, 1991) and its translations, but parsed with different grammars, an action initiated by the Nordic Treebank Network (Nivre et al., 2005). The parallel Sofie treebanks were collected, catalogued and aligned in cooperation with the META-NORD project.[4].

According to the user's choices, a list of treebanks is presented. The resulting list is the intersection between the values for the three criteria; however, each criterion allows multiple values of which the union is taken. For instance, choosing both the NorGram and Sofie collections selects all treebanks from both collections. This is illustrated in Figure 1 where these chosen NorGram and Sofie collections are in boldface. Furthermore, in this example Norwegian Bokmål is selected as the language, which means that only Norwegian treebanks are chosen from these collections. The chosen annotation type in this case was *All*. Because all treebanks chosen in this way are of type *lfg* or *constituency*, these are also automatically marked in boldface.

In the future, it will also be possible to select treebanks based on metadata attributes such as owner, licensing conditions, etc.

4. Visualization

Visualization is a nontrivial need for the exploration of highly detailed treebanks. Once a treebank is selected, its sentences are listed on the Sentence Overview page. Clicking on a sentence shows its structure by means of visualizations dependent on the annotation formalism as well as user preferences. For instance, the same German sentence from the Sofie constituency treebank can be visualized with a traditional tree structure as in Figure 2 or with Tiger-style horizontal and vertical lines as in Figure 3.

Structures in the LFG formalism are well supported through the integration of the LFG Parsebanker tool (Rosén et al., 2009), originally developed in the TREPIL project for the construction and exploration of LFG parsebanks. LFG treebanks are highly detailed and contain several levels of representation such as c-structures (constituent structures) and f-structures (functional structures, feature-value matrices). These structures are juxtaposed in the interface, with mouse-over highlighting to indicate corresponding elements in both structures. This is illustrated in Figure 4, where placing the cursor at PROPP in the c-structure causes highlighting of the value of the TOPIC feature in the (simplified) f-structure.

Parallel treebanks are visualized by displaying structures for aligned sentences in different languages next to each other. In Figure 5, German and Swedish constituency structures

[2] AllegoServe, http://allegroserve.sourceforge.net/.

[3] http://www.hf.uio.no/ifikk/english/research/projects/proiel//

[4] http://www.meta-nord.eu, under the umbrella of META-NET and linked to META-SHARE

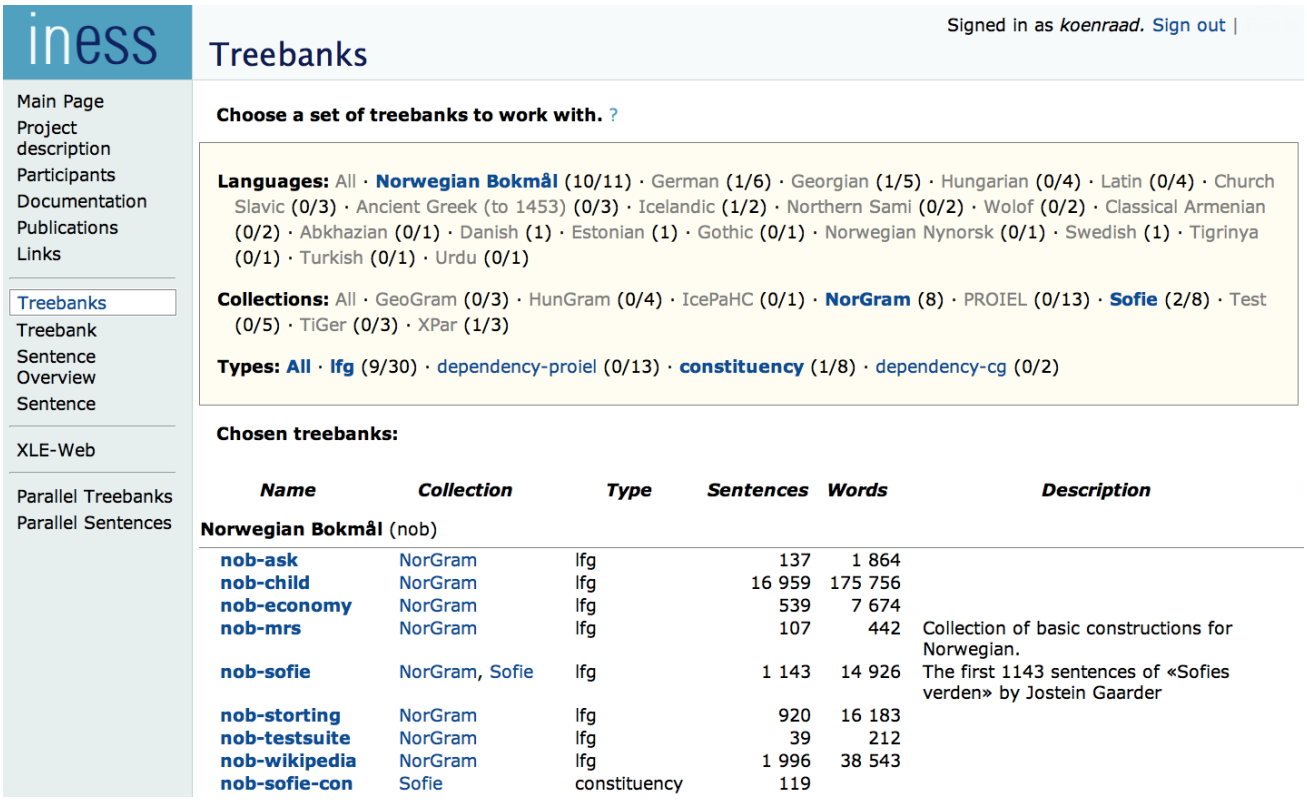

Figure 1: Screenshot of the treebanks selection interface

are juxtaposed. Phrase-aligned treebanks are also catered for, thanks to a methodology developed in the XPAR project (Dyvik et al., 2009). An example of phrase-aligned c- and f-structures in Georgian and Norwegian is given in Figure 6.

5. INESS-Search

Powerful tools for interactively searching and filtering treebanks are of primary importance in a treebanking infrastructure. The query syntax should be expressive in order to cater to a variety of research needs and the implementation should be efficient for a fast turnaround when searching a very large treebank. Furthermore, a search tool should be as simple and uniform as possible across different annotation types.

Currently a number of corpus search tools support searching and viewing of treebanks, such as CorpusSearch 2,[5] TrED 2.0,[6] and TIGERSearch.[7] They have a query language adapted to syntactic annotation in certain formats. TIGERSearch also includes a graphical query building interface. An overview of treebank query systems can be found elsewhere (Lai and Bird, 2004). Most search and viewing tools need to be downloaded and installed on the user's machine. TIGERSearch, which is no longer maintained, has been reimplemented as INESS-Search in Common Lisp, its functionality has been expanded, its query language has been made simpler and it has been integrated into the INESS web interface. INESS-Search can be used to query constituency and dependency treebanks, but it also contains extensions which are necessary for querying LFG f-structures, which are directed, possibly cyclic graphs rather than trees. An evaluation of INESS-Search against TIGERSearch and some other treebank search systems based on the TIGER treebank shows that the former is as fast or significantly faster on most types of queries (Meurer, 2012 forthcoming). Whereas the expressive power of TIGERSearch merely equals that of the existential fragment of first-order predicate logic over node variables (all node variables are implicitly existentially quantified), INESS-Search implements full first-order predicate logic. Its implementation in Common Lisp is seamlessly integrated in the infrastructure and it can therefore be used via a web interface in a straightforward way. This tight integration also means that search can be dynamic, i.e. changes in the treebank are immediately accessible to the search mechanism. This is particularly useful during the construction phase of the treebank when changes are frequent.

A graphical query construction tool has not been developed for INESS-Search since the query syntax is compact and intuitive enough (after some practice) to make such a device unnecessary. Moreover, it would be difficult to expose the full query syntax (including negation, disjunctions and quantifier scoping) in an elegant, easy-to-use graphical tool; implementing only the easier parts of the query syntax (like the existential fragment) would unneccessarily restrict the user to a less expressive subset of the language.

A mechanism for displaying and exporting search results in

[5] http://corpussearch.sourceforge.net/

[6] http://ufal.mff.cuni.cz/tred/

[7] http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/oldindex.shtml (Lezius, 2002)

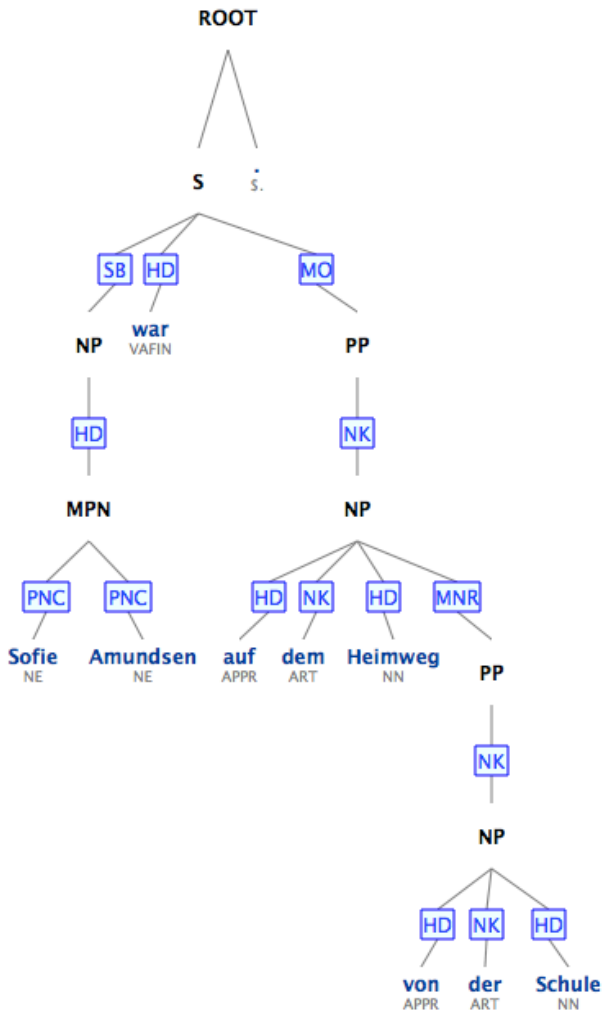

Figure 2: Screenshot of a constituent visualization, traditional branches

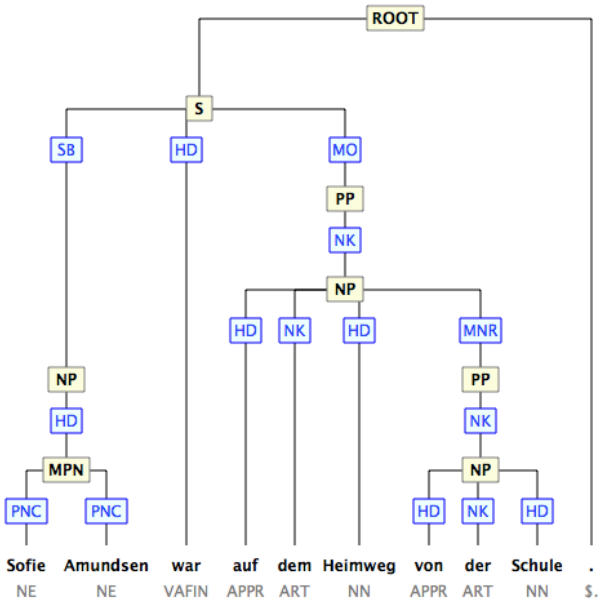

Figure 3: Screenshot of a constituent visualization, TIGER style branches

a flexible way is currently under development.
We may consider a few query examples. If one wants to find all sentences containing NPs that immediately dominate APs, the TIGERSearch expression in 1 can be used to that effect.

(1) [cat ="NP"] > [cat = "AP"]

In INESS-Search one may also use query 1, but the abbreviated syntax in 2 has the same results.

(2) NP > AP

If one wants to find all sentences containing NPs that immediately dominate APs that in turn immediately dominate PPs, variables are needed in TIGERSearch, as illustrated in 3.

(3) [cat = "NP"] > #x:[cat = "AP"] & #x > [cat = "PP"]

In INESS-Search the simplified expression in 4 has the same results as 3.

(4) NP > AP > PP

One result from this search in the Tiger treebank is shown in Figure 7, where the categories in the search expression are highlighted in red.
The query intentions in the examples in 5 are not expressible in TIGERSearch due to the lack of universal quantification.[8]

(5) Q2: Find sentences that do not include the word "saw".
Q5: Find the first common ancestor of sequences of a noun phrase followed by a verb phrase.

The examples in 6 are INESS-Search queries expressing the intentions in the examples in 5.

(6) Q2: !(#x:"saw" = #x)
Q5: #c >∗ #n:NP !>∗ #v & #c >∗ #v:VP !>∗ #n
& !(#c >∗ #x >∗ #n & #x >∗ #v & #n .∗ #v)

As a convention, variables like c, v, and n in example 6 Q5 that occur in positive contexts are treated as existentially quantified, whereas variables like x in Q2 and Q5 that only occur in negated contexts are taken to be universally quantified and in the scope of all existential quantifiers.
A variable occuring in a positive context can be explicitly marked as universally quantified by using '%' as the variable marker instead of '#'. In case the intended quantifier scoping deviates from the default, the scoping order can be given explicitly, as illustrated in query example 7 to search for all sentences where each NP dominates an N:

(7) (%x #y): %x:NP > #y:N

[8] Queries Q2 and Q5 are taken from Lai and Bird´s survey of treebank query systems (Lai and Bird, 2004), where they list typical queries that a query system should be able to express. Admittedly, the results of Q2 would be rather uninteresting to many.

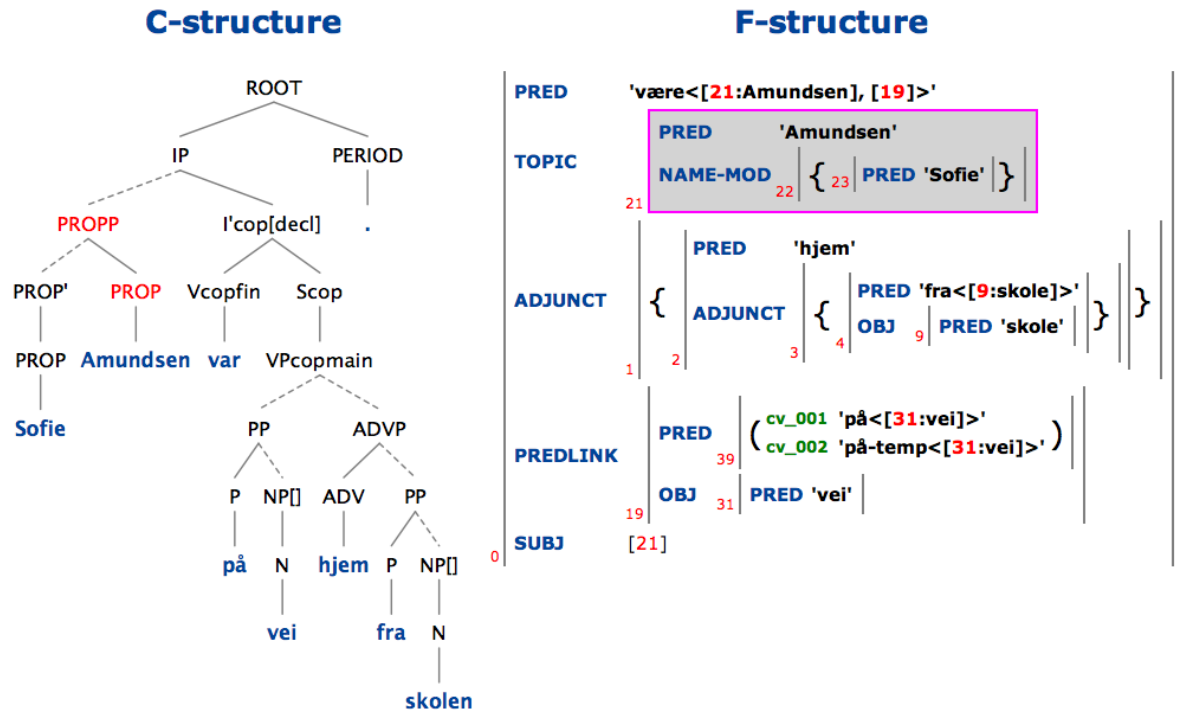

Figure 4: LFG c-structure and corresponding f-structure with mouse-over highlighting

Several operators have been implemented that allow for conveniently querying more complex tree node or f-structure constellations. A rule operator can be used for specifying parent–children constellations (e.g., 8). Operators specifically tailored to LFG structures are a projection operator, an extended-head operator, and regular expressions over f-structure attributes (e.g., 9).

(8) #c → AP .∗ PP

(9) #f >(TOPIC & XCOMP* OBJ) #g

In addition, functionality for querying parallel treebanks (Dyvik et al., 2009) is being further developed.

6. Interactive annotation

INESS offers advanced tools for the online interactive construction of treebanks, in particular LFG treebanks. The LFG Parsebanker tool (Rosén et al., 2009) was developed for this purpose. It was inspired by the [incr tsdb()] environment (Bender et al., 2011), a further development of the TSNLP methodology (Oepen and Flickinger, 1998), which supports annotation and grammar development through test suite management and regression testing. Our approach to treebanking has much in common with the approach advocated there, with the parsed corpus itself constituting part of the grammar development tool. But whereas their approach is mostly applied to HPSG grammars, ours is specialized for LFG grammars.

The LFG Parsebanker has been fully integrated into the infrastructure and offers the following workflow:

- A corpus is batch parsed with XLE (Maxwell and Kaplan, 1993; Kaplan et al., 2002) and all analyses (packed) are stored.
- For each sentence, discriminants are computed (Rosén et al., 2007) and presented to the annotator for disambiguation, as illustrated in Figure 8.
- The annotator's choice of discriminants is applied to the parse result and the remaining structures are displayed. This process can be repeated until the sentence is disambiguated.
- The chosen discriminants are stored; they can be undone by the annotator or automatically reapplied after reparsing.

Furthermore, a system for comments and issue tracking is provided to further assist in grammar development. Statistics are kept to measure discriminant frequencies and inter-annotator agreement. An integrated web-based parsing platform, the XLE-Web interface (Rosén et al., 2005), allows interactive parsing of sentences entered by the user. The infrastructure thus offers a complete online environment for the construction of LFG treebanks, without the need to download and install any software. This setup is currently being used to construct a number of LFG treebanks online for different languages, including a large Norwegian treebank.

7. Conclusion and future work

Treebanks are potentially highly useful, but high quality treebanks are very expensive to construct. Therefore long term archiving, curation and dissemination of these resources needs attention in order to maximize their exploitation in R&D. On the one hand, many treebanking projects produce very useful results but their dissemination is often limited to a particular treebank or type of treebank. On

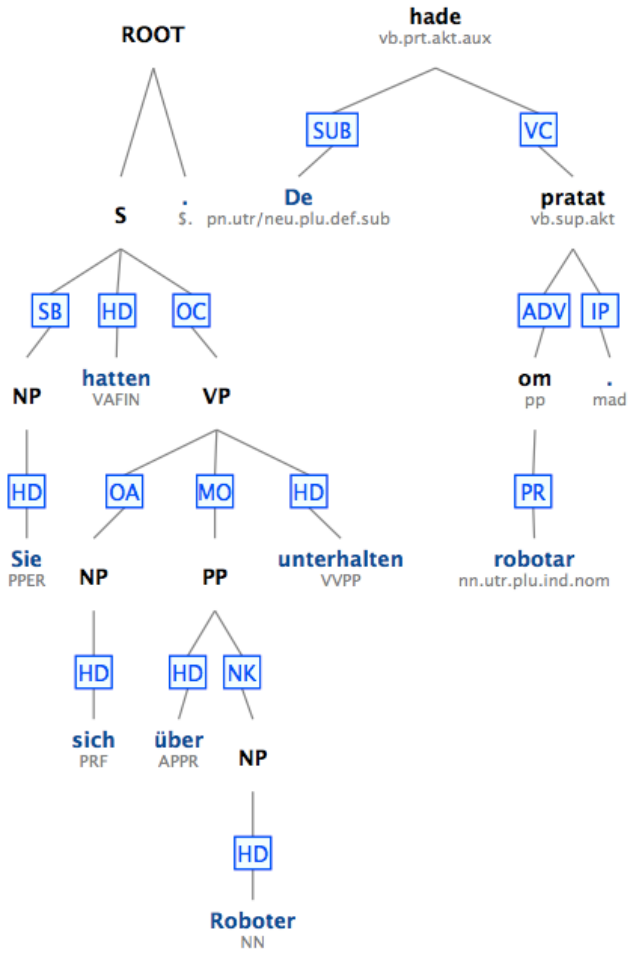

Figure 5: Parallel display of German constituency and Swedish dependency structures: *Sie hatten sich über Roboter unterhalten.* / *De hade pratat om robotar.*

the other hand, current open infrastructures for language resources and tools, such as META-SHARE and CLARIN, aim at building large catalogs and target large user groups, but they do not offer specific middleware or services needed for the construction and exploration of treebanks. INESS intends to fill this gap. We have in this paper presented a status report for this infrastructure.

INESS offers an expanding number of services for a steadily increasing number of treebanks, placing unprecedented emphasis on usability, on powerful search and analysis, and on advanced visualization. INESS intends to be an open infrastructure: it invites the participation of other treebanking projects and is currently negotiating with partners to set up mirrors of the infrastructure. By establishing common access, exploration and visualization of various treebank types through a uniform web-based interface, the threshold for actually using treebanks is lowered for a potentially large audience of users.

Currently, the user base of INESS is still limited. In the past year, the infrastructure has been tested mostly by internal users, and feedback has resulted in several improvements to the user interface. A more extensive, systematic user evaluation is scheduled for 2013. It is our goal that eventually, users with even a minimal linguistic background will consult treebanks in INESS almost as easily as they consult a dictionary or grammar book. We also believe that grammar teaching materials will eventually link to treebanks.

Building on our experiences so far, we envisage that INESS will soon provide web services for many treebanks, including a large treebank for Norwegian with unprecedented detail which is presently being constructed. As the infrastructure is scaling up, syntactic analysis and search must run on high-performance computing platforms in order to have an acceptable turnaround, especially when reparsing (and redisambiguating) an entire corpus with a new grammar version. Furthermore, treebanks need considerably more storage space than corpora annotated at word level only, especially when all analyses of each sentence are stored and discriminants are cached. The INESS infrastructure therefore runs on a 128-core HPC cluster using fast disk access and high-speed internal networking (cf. Figure 9). In its next phase, it will also use national eInfrastructure facilities.

Although the INESS infrastructure is fully operative, further research will allow its evolution in response to new requirements and technologies. Areas of special attention are the search and visualization middleware and the interface and user profiles. In particular, visualization of large structures is a daunting challenge. On the one hand, large screens with high resolutions are desirable physical media. On the other hand, there will be a need for continued research on innovative visualization, which may draw on experiences with visualization of large, complex structures in other fields.[9]

Access to the INESS resources and services is as yet on the basis of ad hoc usernames, while some treebanks are fully open. Current work is aimed at improved handling of licenses and metadata (in cooperation with META-NORD and CLARIN) and the integration of federated authentication and authorization, allowing users from several affiliations to log in with their local user name. Tests of federated user authentication were recently successfully concluded, using an interface to the Norwegian national FEIDE federated ID provider through a SAML 2.0 protocol (Uninett, 2010). Once a trust mechanism for authenticating users from other locations is in place, users will be able to create profiles, define preferences and store search expressions for future use. Users will also be able to upload and parse 'private' treebanks, even if sharing of treebanks is highly encouraged.

8. Acknowledgments

The research reported on in this paper has received funding from the Research Council of Norway under the National Financing Initiative for Research Infrastructure and from the ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, grant agreement no. 270899.

[9] E.g. Jmol for visualization of large molecules, http://www.jmol.org/.

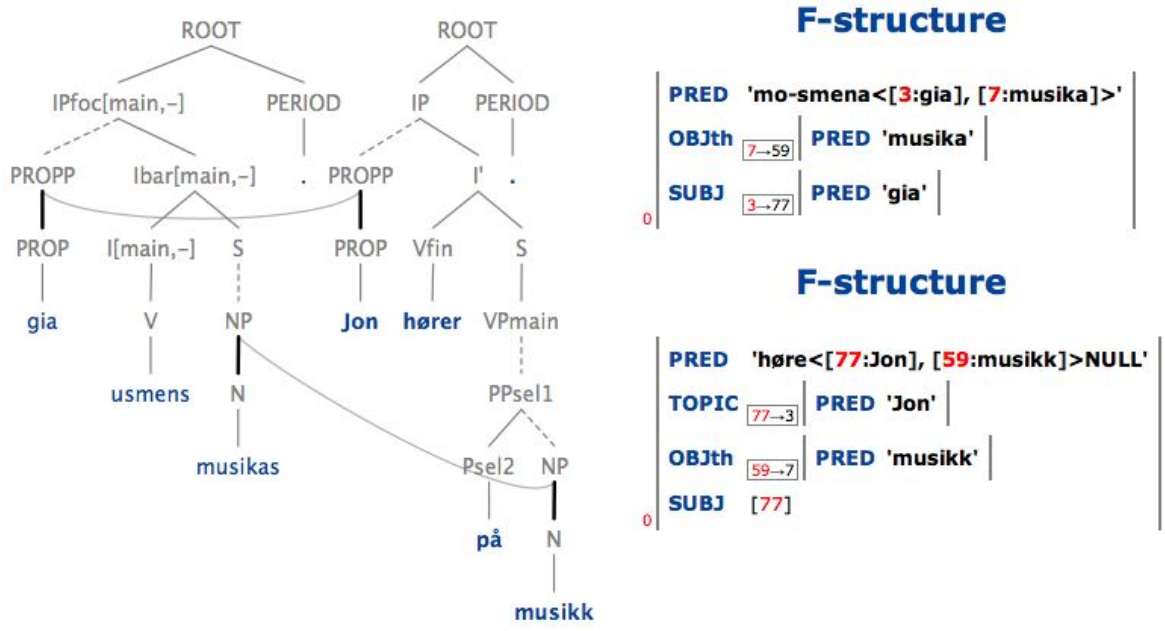

Figure 6: Parallel display of Georgian and Norwegian c-structures and f-structures

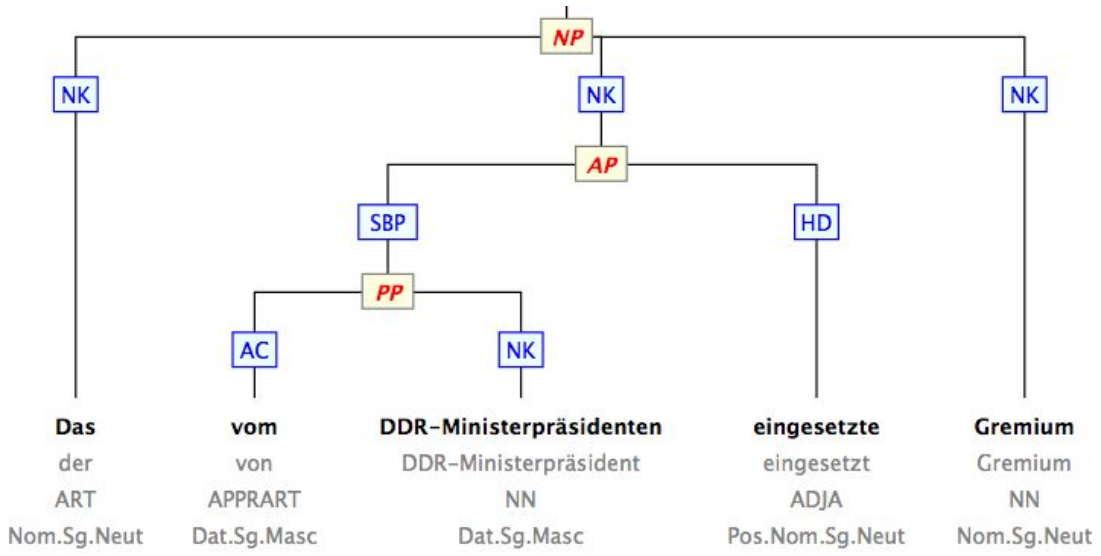

Figure 7: Search example: a solution for NP > AP > PP

F-structure discriminants | show all

7:13	'lieben<[],[]>' OBJ 'Maria'	1	compl (1)
7:1	'lieben<[],[]>' OBJ 'Peter'	1	compl (1)
7:13	'lieben<[],[]>' SUBJ 'Maria'	1	compl (1)
7:1	'lieben<[],[]>' SUBJ 'Peter'	1	compl (1)

Figure 8: Discriminants for the ambiguous German sentence *Peter liebt Maria*.

9. References

Emily M. Bender, Dan Flickinger, and Stephan Oepen. 2011. Grammar engineering and linguistic hypothesis testing: Computational support for complexity in syntactic analysis. In Emily M. Bender and Jennifer E. Arnold, editors, *Language from a Cognitive Perspective: Grammar, Usage, and Processing*, CSLI Lecture Notes 201, pages 5–29. CSLI Publications.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The Prague Dependency Treebank. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, chapter 7, pages 103–127. Kluwer Academic Publishers.

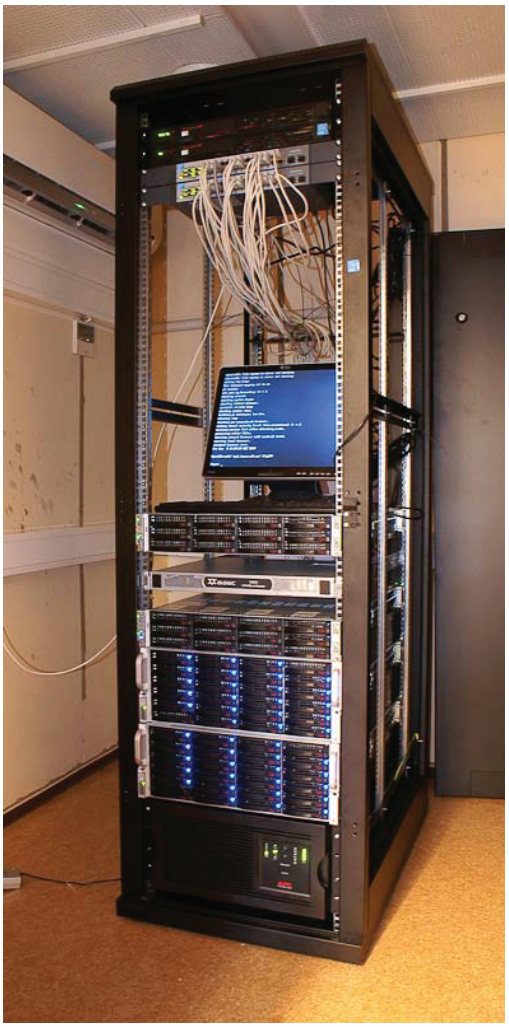

Figure 9: The INESS HPC cluster

Helge Dyvik, Paul Meurer, Victoria Rosén, and Koenraad De Smedt. 2009. Linguistically motivated parallel parsebanks. In Marco Passarotti, Adam Przepiórkowski, Sabine Raynaud, and Frank Van Eynde, editors, *Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories*, pages 71–82, Milan, Italy. EDUCatt.

Jostein Gaarder. 1991. *Sofies verden: roman om filosofiens historie*. Aschehoug, Oslo, Norway.

Jan Hajič. 1998. Building a syntactically annotated corpus: The Prague Dependency Treebank. In *Issues of Valency and Meaning*, pages 106–132. Karolinum, Praha.

Ronald M. Kaplan, Tracy Holloway King, and John T. Maxwell. 2002. Adapting existing grammars: the XLE experience. In *Proceedings of the COLING-2002 Workshop on Grammar Engineering and Evaluation, Taipei, Taiwan*.

Catherine Lai and Steven Bird. 2004. Querying and updating treebanks: A critical survey and requirements analysis. In *Proceedings of the Australasian Language Technology Workshop*, pages 139–146.

Wolfgang Lezius. 2002. TIGERSearch – Ein Suchwerkzeug für Baumbanken. In Stephan Busemann, editor, *Proceedings der 6. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2002), Saarbrücken*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

John Maxwell and Ronald M. Kaplan. 1993. The interface between phrasal and functional constraints. *Computational Linguistics*, 19(4):571–589.

Paul Meurer. 2012 (forthcoming). INESS-Search: A search system for LFG (and other) treebanks. In *Proceedings of the LFG '12 Conference*.

Joakim Nivre, Koenraad De Smedt, and Martin Volk. 2005. Treebanking in Northern Europe: A white paper. In Henrik Holmboe, editor, *Nordisk Sprogteknologi 2004. Årbog for Nordisk Sprogteknologisk Forskningsprogram 2000-2004*, pages 97–112. Museum Tusculanums Forlag, Copenhagen.

Stephan Oepen and Daniel P. Flickinger. 1998. Towards systematic grammar profiling. Test suite technology ten years after. *Journal of Computer Speech and Language*, 12(4):411–436.

Victoria Rosén, Paul Meurer, and Koenraad De Smedt. 2005. Constructing a parsed corpus with a large LFG grammar. In *Proceedings of LFG'05*, pages 371–387. CSLI Publications.

Victoria Rosén, Paul Meurer, and Koenraad De Smedt. 2007. Designing and implementing discriminants for LFG grammars. In Tracy Holloway King and Miriam Butt, editors, *The Proceedings of the LFG '07 Conference*, pages 397–417. CSLI Publications, Stanford.

Victoria Rosén, Paul Meurer, and Koenraad De Smedt. 2009. LFG Parsebanker: A toolkit for building and searching a treebank as a parsed corpus. In Frank Van Eynde, Anette Frank, Gertjan van Noord, and Koenraad De Smedt, editors, *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories (TLT7)*, pages 127–133, Utrecht. LOT.

Uninett. 2010. Feide integration guide: Integrating a service provider with Feide. Technical Report version 1.3, Uninett, Trondheim, August.

Gertjan van Noord. 2009. Huge parsed corpora in LASSY. In Frank Van Eynde, Anette Frank, Koenraad De Smedt, and Gertjan van Noord, editors, *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories (TLT7)*, pages 115–126. LOT.

Joel Wallenberg, Anton Karl Ingason, Einar Freyr Sigurðsson, and Eiríkur Rögnvaldsson. 2011. Icelandic Parsed Historical Corpus (IcePaHC) version 0.9.