Chapter 25

LFG treebanks

Victoria Rosén
University of Bergen

Treebanks are syntactically annotated corpora. LFG treebanks are collections of LFG analyses, usually created by parsing a corpus with an LFG grammar. This chapter provides an overview of existing LFG treebanks and explains how they are created, how they may be searched, and what their potential use may be to the LFG and other communities.

1 Introduction

Annotated corpora are important resources for many branches of linguistics, language studies and natural language processing. A common form of corpus annotation consists of labeling words with their parts of speech, lemmas and morphosyntactic features, such as number, person, tense, etc. Using only annotation at the word level limits the potential to search for important grammatical information, such as syntactic constructions, grammatical functions and predicate–argument relations. The usefulness of corpora is therefore greatly enhanced if they also include syntactic annotation, such as phrase structure and functional relations. Syntactically annotated corpora are usually called treebanks; if they are created by parsing, they may also be called parsed corpora or parsebanks.

LFG treebanks are treebanks annotated according to the LFG formalism. They are usually created as parsebanks, by parsing a corpus with an LFG grammar and disambiguating the parse results. An LFG parsebank is thus essentially a collection of analyses according to a grammar. LFG parsebanks encode a wealth of morphological, syntactic and semantic information in their c- and f-structure representations, and tend to be more detailed than treebanks adhering to other

Victoria Rosén. 2023. LFG treebanks. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*, 1169–1206. Berlin: Language Science Press. DOI: 10.5281/zenodo.10185992

formalisms. The term treebank is well established even if the treebank may contain f-structures, which are directed graphs rather than trees.

This chapter is aimed at two audiences. The first target group consists of linguists who may wish to learn to use LFG treebanks in order to find data for their research. The second target group is linguists who may wish to build LFG treebanks as part of a grammar development project.

A major platform for LFG treebanking is INESS (Infrastructure for the Exploration of Syntax and Semantics) at the CLARINO Bergen Center (University of Bergen, Norway).[1] This infrastructure will be further introduced below and will be used throughout the chapter to illustrate the various possibilities of LFG treebanking.

Section 2 describes how LFG treebanks can be created through parsing with the Xerox Linguistic Environment and further processed with the LFG Parsebanker. In Section 3 the LFG treebanks in the INESS treebanking infrastructure are presented. Section 4 demonstrates how LFG treebanks may be queried with INESS Search. Finally, Section 5 describes approaches to conversion between LFG treebanks and treebanks adhering to other formalisms.

2 Building LFG treebanks

2.1 Basic requirements

A parser, an implemented grammar and lexicon, and efficient disambiguation tools are prerequisites for creating a parsebank. A useful set of tools in this respect is the Xerox Linguistic Environment (XLE), developed at the Palo Alto Research Center and the Xerox Research Centre Europe in Grenoble. XLE includes both a parser and a generator for LFG grammars, and it is suitable for grammar implementation on a small or large scale (Crouch et al. 2011, Maxwell & Kaplan 1993). For detailed information on XLE, see Forst & King 2023 [this volume].

A grammar and lexicon with wide coverage are essential for building a large treebank of authentic texts, as well as for other applications. Grammar development is however a process which typically starts with a small set of rules which is successively expanded. In this development, the grammar must constantly be tested to see whether all the old rules still work in addition to the new rules. In this incremental process, a corpus, even a small one initially, may be useful as a

[1] https://clarino.uib.no/iness. INESS was built in the eponymous project (2010-2017) with funding from the Research Council of Norway and the University of Bergen (Rosén et al. 2012, Meurer et al. 2013).

test suite for parsing. As the grammar grows, it can be tested on a larger corpus and further improved. Larger grammars and lexicons do however increase the ambiguity in the analyses, so that efficient disambiguation is important.

XLE-Web[2] is a web-based implementation of XLE that was first developed in the LOGON and TREPIL projects (Rosén et al. 2005, 2006). XLE-Web uses the same parsing technology and software as XLE, but differs from the original platform in several ways. The original XLE is a standalone, integrated platform for grammar writing and debugging, whereas XLE-Web can be used through any modern browser. XLE-Web does not have tools for grammar writing, but it offers excellent tools for disambiguation.

As mentioned above, ambiguity becomes a considerable problem as the grammar grows. Therefore, XLE-Web offers *discriminant disambiguation* to efficiently select the intended analysis among possibly many alternative analyses. Discriminant analysis is a technique for identifying minimal differences between analyses and letting disambiguation proceed by resolving these differences rather than by inspecting whole structures (Rosén et al. 2007). An example of the XLE-Web display with discriminants is provided in Figure 1 for the ambiguous sentence *He saw the girl with binoculars*,[3] parsed with the English ParGram grammar.[4]

This sentence has two possible analyses due to a PP attachment ambiguity: *with binoculars* may be either an ADJUNCT of the clause or an ADJUNCT in the OBJ. Whereas XLE offers packed f-structures, XLE-Web offers packed representations for both c- and f-structures. A packed representation presents all analyses in one graph, with indices at choice points. In the middle of Figure 1 is a packed c-structure with one choice point which splits into the subtrees labeled *a1* and *a2*. A corresponding choice can be seen in the packed f-structure shown on the right in the figure. Although the disambiguated f-structure will have an ADJUNCT either on the outer level or inside the OBJ, both functions occur in the packed f-structure, labeled with *a1* and *a2* respectively.

On the left in the figure is a table with discriminants computed on the basis of these choice points. They present the user with each individual distinction between the analyses. There are two f-structure discriminants and ten c-structure discriminants.[5] F-structure discriminants describe paths through the f-structure

[2]https://clarino.uib.no/iness/xle-web

[3]In this and many subsequent f-structures, the *PREDS only* mode of display has been chosen. PREDS only mode displays only PRED values and the attribute paths which lead to them. This mode is often preferred when a full f-structure is too large to be easily legible.

[4]This grammar was developed in the Parallel Grammar (ParGram) project, see Section 3.6.2.

[5]In some cases there may also be lexical and morphological discriminants, but not for this sentence, which does not display any lexical ambiguities.

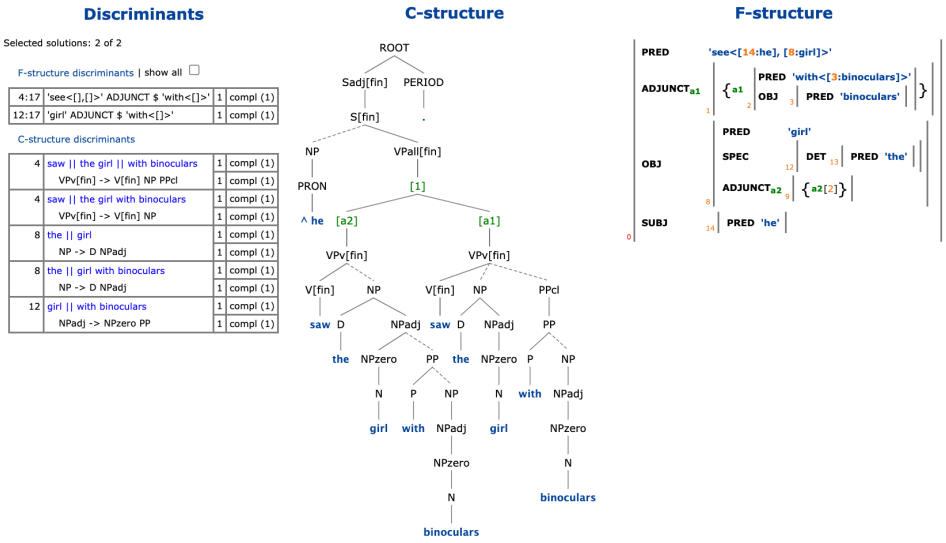


Figure 1: Analysis with discriminants and packed c- and f-structures for *He saw the girl with binoculars.*

from a PRED value to another PRED value or an atomic value. The two f-structure discriminants shown here indicate that the phrase *with binoculars* is an ADJUNCT either of the verb *see* or of the noun *girl*. The ten c-structure discriminants present the various minimal subtrees (a minimal subtree being a mother node and its daughter nodes) that make up the subtrees indexed with *a1* and *a2*. C-structure discriminants are either constituent discriminants, which show the bracketing of a substring, or rule discriminants, which show the labeled bracketing of a substring, expressed as a phrase structure rule. Rule discriminants are always displayed directly under the corresponding constituent discriminant, thus showing clearly which string of words the rule represents a bracketing of.

A discriminant may be chosen by clicking on it, or rejected by clicking on *compl* (for *complement*).[6] After a discriminant or its complement has been clicked on, it is displayed in boldface; the choice may be reversed by clicking on the boldfaced discriminant, thus resetting it. Since there are only two analyses for the

[6]The numbers to the left of the discriminants are anchors, which are necessary in case the same word or phrase occurs more than once in the sentence. In c-structure discriminants the anchor identifies the position of the first character in the substring. In f-structure discriminants the anchors identify the position of the first character of the words that project the PRED values in the discriminant. The number to the right of a discriminant (or its complement) indicates the number of solutions that will remain after the discriminant (or its complement) is chosen.

sentence in Figure 1, the intended one may be selected by choosing or rejecting any one discriminant. Figure 2 shows the effect of choosing the analysis in which *with binoculars* is an adjunct of the verb *see* by clicking on the first f-structure discriminant, resulting in full disambiguation. Discriminants that have not been chosen and that are no longer relevant for disambiguation, because they do not distinguish between any remaining analyses, are not displayed. This is important for efficiency, since the disambiguator then has fewer discriminants to take into consideration.

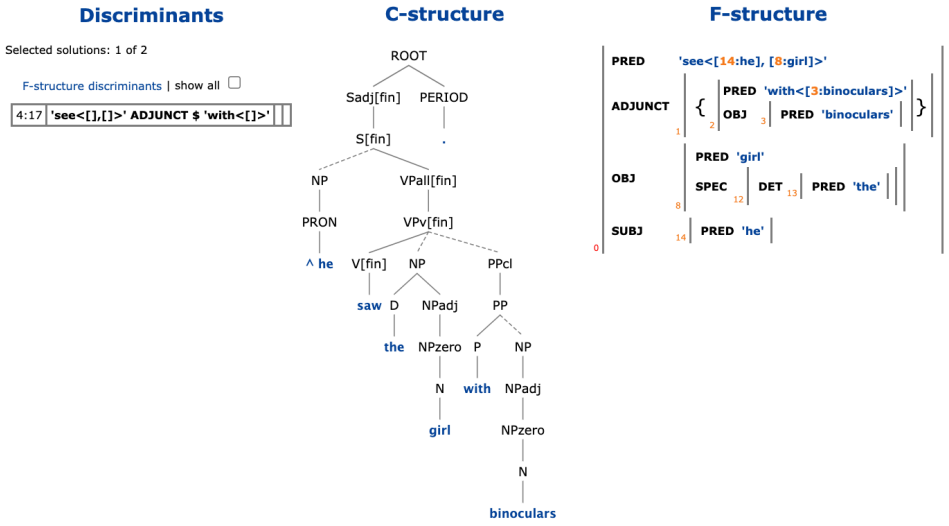


Figure 2: Fully disambiguated analysis for *He saw the girl with binoculars.*

This process may seem like overkill for this simple example which has only two readings. It becomes rewarding, however, when there are multiple ambiguities in the sentence. Even when the combination of ambiguities may give rise to a very large number of analyses, the number of discriminants does not necessarily increase as much, so that discriminant analysis remains comparatively efficient. A more detailed presentation of disambiguation with discriminants in LFG may be found in Rosén et al. (2007).

At the time of writing, the XLE-Web instance at INESS offers online parsing with the ParGram grammars of the following languages: English, French, Georgian, German, Indonesian, Italian, Malagasy, Norwegian, Polish, Tamil, Tigrinya, Turkish, Urdu and Wolof. Some of these have broad coverage, while others are more limited in scope.

2.2 The LFG Parsebanker

The LFG Parsebanker, available in INESS, is an integrated set of tools for creating and searching LFG treebanks (Rosén et al. 2009). It allows texts to be batch parsed with the XLE parser, and it stores the analyses in a database. The resulting parsebank may be disambiguated by using discriminants in the same way as described above. The LFG Parsebanker stores both the analyses and all discriminant choices that were made. This means that the grammar and lexicon may be further developed, and the treebank subsequently reparsed and at least partially redisambiguated with the stored discriminant choices. This method makes it possible to develop the grammar and the treebank in tandem, thus incrementally improving the quality of the analyses. The stored discriminants may also be used for stochastic parseranking. In this way larger parsebanks can be automatically disambiguated.

A possible drawback of constructing a treebank by parsing with an LFG (or other) broad-coverage unification grammar is that the grammar cannot hope to have full coverage for all authentically occurring sentences in a large corpus. Nevertheless, some traditional treebanks that are (at least partially) manually annotated are meant to assign an analysis to every sentence, and a variety of methods are utilized to achieve this. When a sentence is not covered by the grammar, an annotator can, for instance, manually construct an analysis to "fix" the problem. Although this provides an analysis for the treebank, it does not provide an analysis that is consistent with a grammar, and sentences that are not actually grammatical may receive analyses as if they were. In contrast, a pure parsebank does not resort to such ad hoc fixes, since it is often primarily meant to test the coverage and precision of a grammar, so that it is desirable to keep the treebank in sync with the grammar. The LFG Parsebanker therefore does not permit disambiguators to edit the automatically derived analyses, but allows them to make notes for grammar and lexicon development to solve coverage problems.

3 LFG treebanks in INESS

INESS is a treebanking infrastructure for building, hosting and exploring treebanks. It includes the above-mentioned XLE-Web and the LFG Parsebanker. It also has an elaborate infrastructure for browsing, search and visualization, as will be explained below.

INESS accommodates not only LFG treebanks, but also treebanks based on other frameworks, such as HPSG (Pollard & Sag 1994), constituency, and dependency treebanks. The infrastructure makes treebanks available online in an in-

ternet browser, eliminating the need to download treebanks and software for viewing and searching them, thus considerably facilitating access to them. Since INESS hosts many treebanks, there is an interface for treebank selection, as described in Section 3.1.

While some treebanks have completely open access, others require user authentication and authorization. Treebank owners decide under what licensing terms their treebanks are to be made available; some treebanks have restrictive licenses due to copyright of the input texts. The most open license that copyright will allow is recommended (Rosén & De Smedt 2022). INESS participates in the CLARIN Service Provider Federation (SPF), which allows researchers to authenticate themselves by logging in with their own university credentials, thus gaining access to many more treebanks than are freely available. The CLARIN SPF has participant institutions in many countries, both in Europe and beyond. Users not belonging to one of these institutions can apply for a user name and password at CLARIN.[7]

INESS hosts LFG treebanks of varying sizes. The larger treebanks TIGER, the LFG Structure Bank for Polish, and NorGramBank are presented in Section 3.2, Section 3.3 and Section 3.4, respectively. The smaller treebanks are presented in Section 3.5. INESS also hosts several parallel treebanks with LFG annotations, presented in Section 3.6. The INESS interface is described in more detail by Meurer et al. (2020).

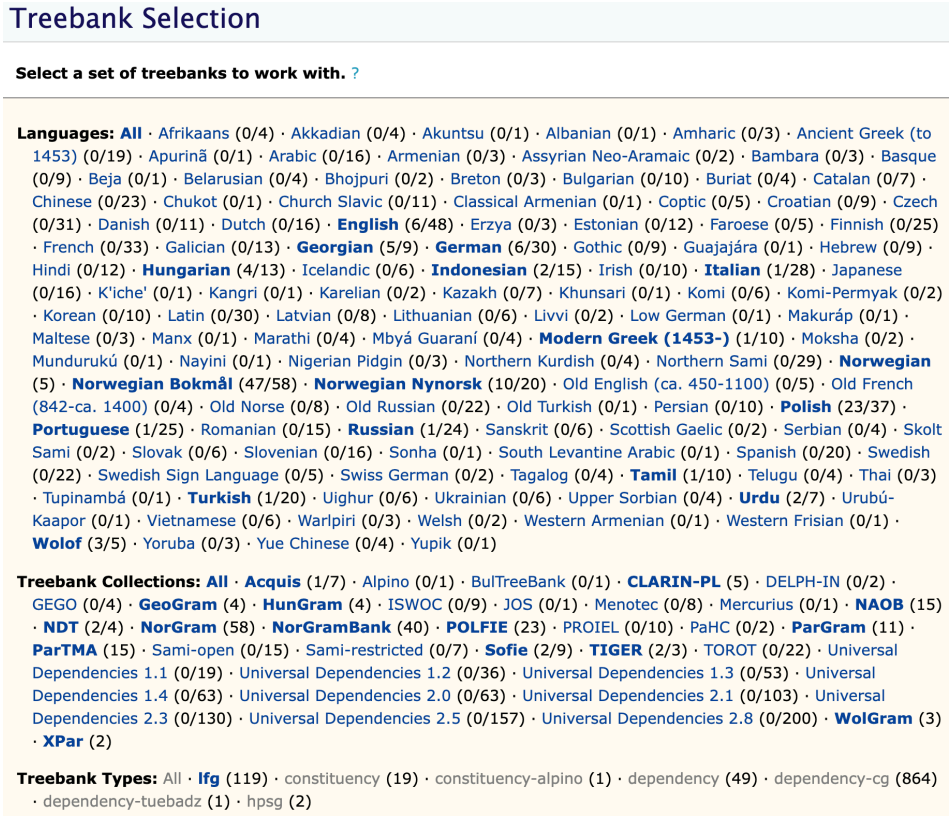
3.1 Selecting treebanks in INESS

The first step in exploring treebanks involves selecting one or more treebanks. At the time of writing, INESS hosts 433[8] treebanks for 115[9] languages. The *Treebank Selection* page in INESS, shown in Figure 3, groups treebanks according to language, collection and type.

[7] CLARIN is a digital infrastructure offering data, tools and services to support research based on language resources (http://clarin.eu).

[8] According to Figure 3, there are 1057 treebanks in total, but this number includes all of the versions of the UD treebanks. If we only count the number of treebanks in Universal Dependencies 2.5 (200), the total number of treebanks is 433.

[9] There are 117 language names, but three of these are Norwegian, Norwegian Bokmål, and Norwegian Nynorsk, and these have been counted as one language: Norwegian. Norwegian Bokmål and Norwegian Nynorsk are the two written standards for the Norwegian language, with a good deal of lexical variation and many differences in spelling and morphology. Most treebank texts are written consistently in one variety or the other, so that users can choose which written variety to explore. Some texts, however, contain both varieties, for instance the proceedings of the Norwegian parliament 'Stortinget'; the latter are categorized simply as Norwegian.



Treebank Selection

Select a set of treebanks to work with. ?

Languages: **All** · Afrikaans (0/4) · Akkadian (0/4) · Akuntsu (0/1) · Albanian (0/1) · Amharic (0/3) · Ancient Greek (to 1453) (0/19) · Apurinã (0/1) · Arabic (0/16) · Armenian (0/3) · Assyrian Neo-Aramaic (0/2) · Bambara (0/3) · Basque (0/9) · Beja (0/1) · Belarusian (0/4) · Bhojpuri (0/2) · Breton (0/3) · Bulgarian (0/10) · Buriat (0/4) · Catalan (0/7) · Chinese (0/23) · Chukot (0/1) · Church Slavic (0/11) · Classical Armenian (0/1) · Coptic (0/5) · Croatian (0/9) · Czech (0/31) · Danish (0/11) · Dutch (0/16) · **English** (6/48) · Erzya (0/3) · Estonian (0/12) · Faroese (0/5) · Finnish (0/25) · French (0/33) · Galician (0/13) · **Georgian** (5/9) · **German** (6/30) · Gothic (0/9) · Guajajára (0/1) · Hebrew (0/9) · Hindi (0/12) · **Hungarian** (4/13) · Icelandic (0/6) · **Indonesian** (2/15) · Irish (0/10) · **Italian** (1/28) · Japanese (0/16) · K'iche' (0/1) · Kangri (0/1) · Karelian (0/2) · Kazakh (0/7) · Khunsari (0/1) · Komi (0/6) · Komi-Permyak (0/2) · Korean (0/10) · Latin (0/30) · Latvian (0/8) · Lithuanian (0/6) · Livvi (0/2) · Low German (0/1) · Makuráp (0/1) · Maltese (0/3) · Manx (0/1) · Marathi (0/4) · Mbyá Guaraní (0/4) · **Modern Greek (1453-)** (1/10) · Moksha (0/2) · Mundurukú (0/1) · Nayini (0/1) · Nigerian Pidgin (0/3) · Northern Kurdish (0/4) · Northern Sami (0/29) · **Norwegian** (5) · **Norwegian Bokmål** (47/58) · **Norwegian Nynorsk** (10/20) · Old English (ca. 450-1100) (0/5) · Old French (842-ca. 1400) (0/4) · Old Norse (0/8) · Old Russian (0/22) · Old Turkish (0/1) · Persian (0/10) · **Polish** (23/37) · **Portuguese** (1/25) · Romanian (0/15) · **Russian** (1/24) · Sanskrit (0/6) · Scottish Gaelic (0/2) · Serbian (0/4) · Skolt Sami (0/2) · Slovak (0/6) · Slovenian (0/16) · Sonha (0/1) · South Levantine Arabic (0/1) · Spanish (0/20) · Swedish (0/22) · Swedish Sign Language (0/5) · Swiss German (0/2) · Tagalog (0/4) · **Tamil** (1/10) · Telugu (0/4) · Thai (0/3) · Tupinambá (0/1) · **Turkish** (1/20) · Uighur (0/6) · Ukrainian (0/6) · Upper Sorbian (0/4) · **Urdu** (2/7) · Urubú-Kaapor (0/1) · Vietnamese (0/6) · Warlpiri (0/3) · Welsh (0/2) · Western Armenian (0/1) · Western Frisian (0/1) · **Wolof** (3/5) · Yoruba (0/3) · Yue Chinese (0/4) · Yupik (0/1)

Treebank Collections: **All** · **Acquis** (1/7) · Alpino (0/1) · BulTreeBank (0/1) · **CLARIN-PL** (5) · DELPH-IN (0/2) · GEGO (0/4) · **GeoGram** (4) · **HunGram** (4) · ISWOC (0/9) · JOS (0/1) · Menotec (0/8) · Mercurius (0/1) · **NAOB** (15) · **NDT** (2/4) · **NorGram** (58) · **NorGramBank** (40) · **POLFIE** (23) · PROIEL (0/10) · PaHC (0/2) · **ParGram** (11) · **ParTMA** (15) · Sami-open (0/15) · Sami-restricted (0/7) · **Sofie** (2/9) · **TIGER** (2/3) · TOROT (0/22) · Universal Dependencies 1.1 (0/19) · Universal Dependencies 1.2 (0/36) · Universal Dependencies 1.3 (0/53) · Universal Dependencies 1.4 (0/63) · Universal Dependencies 2.0 (0/63) · Universal Dependencies 2.1 (0/103) · Universal Dependencies 2.3 (0/130) · Universal Dependencies 2.5 (0/157) · Universal Dependencies 2.8 (0/200) · **WolGram** (3) · **XPar** (2)

Treebank Types: All · **lfg** (119) · constituency (19) · constituency-alpino (1) · dependency (49) · dependency-cg (864) · dependency-tuebadz (1) · hpsg (2)

Figure 3: The INESS user interface for treebank selection, with treebank type *lfg* chosen

A collection contains several treebanks with something in common, for instance that they were developed as part of a specific project, or that they consist of translations of the same text into different languages (including the source language text). A single treebank may belong to more than one collection. Type refers to the annotation type, such as LFG, HPSG, constituency, and dependency, and includes subtypes of these. The user may click on any language, collection or type to make a first choice about which treebanks should be displayed.

In Figure 3 we see the effect of clicking on the type *lfg*; after this choice, only the languages and treebank collections that have LFG treebanks are displayed in boldface. Counting the boldfaced languages in Figure 3 shows that there are 16 languages that have LFG treebanks. After each language name, the numbers in parentheses indicate how many of the treebanks are LFG treebanks; for English, (6/48) means that six of 48 treebanks are LFG treebanks. In a similar manner,

under Treebank Collections, TIGER (2/3) means that two of the three treebanks in the collection called TIGER are LFG treebanks.

Once a first choice has been made by a user, a list of all treebanks matching that choice is displayed. When LFG is chosen, a total of 119 treebanks are listed. The top of this list is shown in Figure 4.[10] For each treebank, this overview shows its name, which collections it belongs to, its annotation type, its size (in sentences and words), whether it has been indexed for search, and the type of license (if any). The user may choose one or more treebanks by ticking off the boxes to the left of the treebank name; clicking on the name of one of the chosen treebanks brings the user to that treebank. When exploring a treebank for the first time, the user is asked to accept the license conditions.

Clicking on a treebank name brings the user to the *Sentence Overview* page for that treebank; the sentences are listed one per line together with information about their disambiguation status. Clicking on a sentence displays the *Sentence* page, where the analysis for that sentence is shown including the textual context the sentence occurs in (the previous and following three sentences).

Click on a treebank name below to proceed. All selected treebanks will be available for viewing and searching. | Show treebank descriptions

Selected all | none	Name	Collection	Type	Sentences	Words	Indexed	License
				17 828 129	252 023 248		
	English (eng)			**533**	**9 501**		
☐	**eng-jrc-acquis** (aligned)		lfg	94	2 188	yes	(Accepted)
☐	**eng-pargram** (aligned)	ParGram	lfg	101	658	yes	Ⓟ CC-BY (Accepted)
☐	**eng-partma**	ParTMA	lfg	45	189	yes	Ⓟ CC-BY (Accepted)
☐	**eng-partma-rat**	ParTMA	lfg	10	163	yes	Ⓟ CC-BY (Accepted)
☐	**eng-partma-scorpion**	ParTMA	lfg	10	127	yes	Ⓟ CC-BY (Accepted)
☐	**eng-partma-tempeval3**	ParTMA	lfg	273	6 176	yes	Ⓟ CC-BY (Accepted)
	Georgian (kat)			**1 242**	**10 719**		
☐	**kat-mrs** (aligned)	GeoGram	lfg	106	374	no	(Accepted)
☐	**kat-pargram** (aligned)	GeoGram, ParGram	lfg	52	231	yes	Ⓟ CC-BY (Accepted)
☐	**kat-partma**	ParTMA	lfg	34	106	yes	Ⓟ CC-BY (Accepted)
☐	**kat-sofie** (aligned)	Sofie, GeoGram	lfg	1 025	9 915	yes	unspecified (Accepted)
☐	**kat-xpar** (aligned)	XPar, GeoGram	lfg	25	93	no	(Accepted)
	German (deu)			**20 278**	**255 589**		
☐	**deu-pargram** (aligned)	ParGram	lfg	102	644	yes	Ⓟ CC-BY (Accepted)
☐	**deu-partma**	ParTMA	lfg	56	262	yes	Ⓟ CC-BY (Accepted)
☐	**deu-partma-manifesto**	ParTMA	lfg	260	3 459	no	Ⓟ CC-BY (Accepted)
☐	**deu-radio**		lfg	1 418	22 952	no	(Accepted)
☐	**deu-tiger**	TIGER	lfg	9 221	114 136	yes	(Accepted)
☐	**deu-tiger/subset**	TIGER	lfg	9 221	114 136	no	(Accepted)

Figure 4: Top of the list of treebanks after the type *lfg* has been chosen

[10] Treebank names in INESS begin with the three-letter ISO 639-3 code for the relevant language.

3.2 The TIGER treebank

The original TIGER treebank of German newspaper text (Brants et al. 2002, 2004) uses a hybrid annotation combining constituency and dependency information; part of it is also annotated with LFG structures. The constituency/dependency part of the treebank was constructed by two different methods. In one method a cascaded probabilistic parser was used in combination with manual annotation with the ANNOTATE tool (Brants & Plaehn 2000). The other method involved parsing with the German LFG grammar, followed by manual disambiguation; the XLE transfer system was employed to change the representations into the TIGER format (Zinsmeister et al. 2002). The LFG analyses were thus originally utilized in an experimental way to construct a more traditional treebank, but they now also constitute a useful resource as a standalone LFG treebank.

Figures 5 and 6 display the constituency/dependency and LFG analyses, respectively, for the sentence in (1). The URLs in parentheses in the captions are PIDs (persistent identifiers). They provide links to the analyses in the treebanks. Such links are persistent as long as the treebank they refer to remains available. For treebanks with certain licensing conditions, the PIDs may only work if the user is logged in and has accepted the license. For LFG treebanks, which are dynamic (they can be reparsed after changes are made to the grammar and/or lexicon), the PIDs are persistent in the sense that they provide a link to the *current* analysis of the sentence in the treebank.

(1) German
Das Angebot ist bereits groß.
the offer is already large
'The offer is already large.'

The tree in Figure 5 contains information about both phrase structure and syntactic functions. The nodes in yellow boxes are phrasal categories, while the nodes in the blue boxes under the S node are syntactic functions: SB for subject, HD for head, MO for modifier and PD for predicate complement.

The c-structure in Figure 6 displays extensive unary branching – many nodes have only single daughters – and many complex category labels, i.e., c-structure nodes subscripted with features enclosed in square brackets. The latter device moves some of the feature complexity of the LFG grammar from the f-structure space into the context-free c-structure space, which improves parsing efficiency while maintaining the simplicity of the c-structure rules. In the f-structure we see that the SUBJ is also analyzed as the TOPIC, the predicate complement is analyzed as an XCOMP-PRED, and the modifier is analyzed as an ADJUNCT.

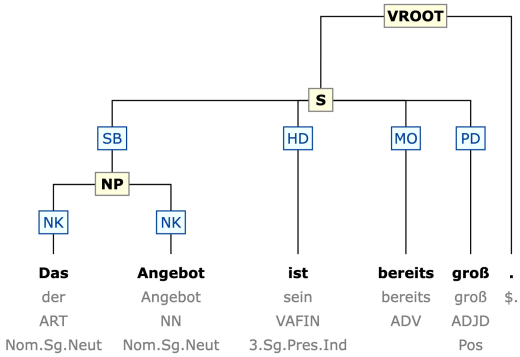


Figure 5: TIGER constituency/dependency analysis of (1) (http://hdl.handle.net/11495/D8B8-3970-851A-3@dep138682)

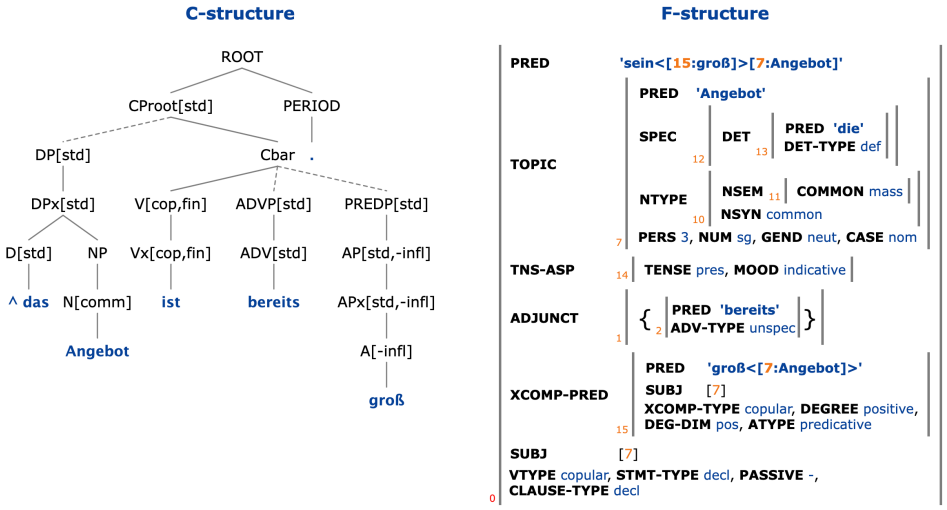


Figure 6: TIGER LFG analysis of (1) (http://hdl.handle.net/11495/D8B8-3970-851A-3@lfg41730)

3.3 The LFG Structure Bank for Polish

The LFG Structure Bank for Polish was built by parsing a corpus with the POLFIE grammar (Patejuk & Przepiórkowski 2012, 2014). This grammar was created by reusing context-free grammar rules written for another parser for Polish, Świgra, and adding annotations for building the f-structures. The corpus for the treebank is the one-million word subcorpus of the National Corpus of Polish[11] which has been manually annotated, the same subcorpus that was used for the previously annotated Składnica treebank.[12]

In INESS, the treebanks created by the POLFIE grammar are all in one large collection, also called POLFIE. This collection includes the LFG Structure Bank for Polish as well as other treebanks. The size of the POLFIE collection is 179,994 sentences and 2,022,026 words. Some of the subtreebanks in POLFIE are also in other collections: CLARIN-PL, ParGram and ParTMA.

Sample c- and f-structures from the POLFIE treebank for the sentence in (2) are given in Figure 7.

(2) Polish
Drzewo został-o ścięt-e wczoraj.
tree.NOM.SG.N get.3SG.N cut.NOM.SG.N yesterday
'The tree was cut down yesterday.'

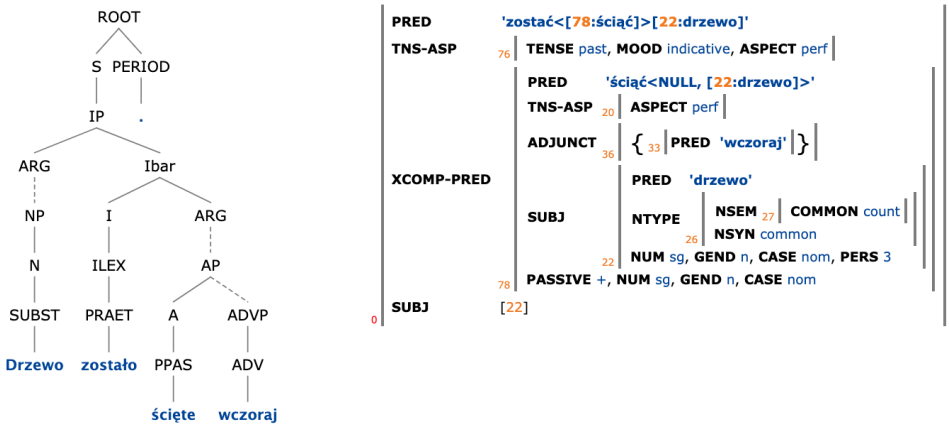


Figure 7: C- and f-structures for the Polish sentence in (2) (http://hdl.handle.net/11495/D8B8-3970-851A-3@lfg1411740)

[11] http://nkjp.pl/index.php?page=0&lang=1
[12] http://zil.ipipan.waw.pl/Składnica

In the c-structure we see some familiar categories such as A, ADV, ADVP, NP, N, I, Ibar, etc., but there are also categories which we might not immediately be able to identify, such as ILEX, PRAET and PPAS. Some terms in the f-structure may also be unfamiliar, such as NTYPE, NSEM and NSYN.[13] Treebank documentation should ideally be made available by treebank creators to assist users in exploring the treebank; unfortunately INESS lacks documentation for many treebanks.

An overview of all *indexed attributes* for each treebank may be found on the *Treebank Details* page. The indexed attributes are all labels used in the treebank annotation that can be searched for. For LFG treebanks, these attributes include *cat* (category) and *edge* (feature or attribute, in more standard LFG terminology). A screenshot of the top of this page is shown in Figure 8.

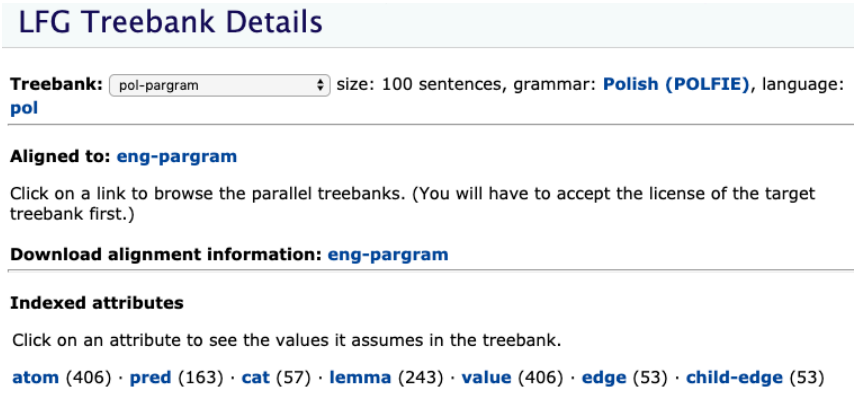


Figure 8: Treebank details for POLFIE

Clicking on *cat* and *edge* under *Indexed attributes* produces the lists in Figure 9. These lists are shown sorted according to frequency; we see that, for instance, the category NP occurs 236 times in this subcorpus (pol-pargram) consisting of 100 sentences.

3.4 NorGramBank: A Norwegian LFG parsebank

The INESS project had the twofold goal of building a treebanking infrastructure and of building the first large treebank for Norwegian. The result of the latter effort is the treebank collection NorGram, consisting of 15 million sentences (215

[13] These f-structure attributes also occur in Figure 6, and they illustrate the parallelism on the f-structure level achieved by the ParGram grammars; see Section 3.6.2.

cat, distinct values: 57

This table shows all values of the attribute, together with their corpus counts.

Sorted (•) by frequency ( ) alphabetically

236	NP	6	INF
201	N	5	MOD
199	SUBST	5	PPAS
156	IP	5	ADVP
115	ARG	5	RM
103	S	4	QUB
98	--	4	XPextr
86	ROOT	4	A-TYPE
83	PERIOD	4	PADJ
63	I	4	NUMP
63	ILEX	4	NUMbare
58	Ibar	4	NUM
56	AP	3	INT-MARK
56	PRAET	3	ADV
54	A	3	PRON
47	ADJ	3	PPRON3
40	AP-SURROUND-OR-NONE	2	SIEBIE
39	FIN	2	PACT
37	COMMA	2	MODPART
28	NEG	2	PADV
24	PP	2	XP_FOCUS
23	P	2	FM
23	PREP	2	CNEG
15	CP	2	CONJ
15	CPbare	1	IMPS
10	COMP	1	PCON
7	PRON-TYPE		
7	PSUBST		
7	IMPT		
7	XPsem		

edge, distinct values: 53

This table shows all values of the attribute, together with their corpus counts.

Sorted (•) by frequency ( ) alphabetically

587	-	12	TYPE
422	PRED	11	COMP
326	CHECK	10	_PREDICATIVE
326	_CAT	10	COMP-FORM
287	CASE	8	PFORM
285	NUM	8	OBL-STR
279	GEND	7	OBJ-TH
226	PERS	6	OBL
206	NTYPE	5	REFLEXIVE
206	NSYN	5	_RQR
206	NSEM	5	XCOMP
206	COMMON	4	OBL-LOCAT
132	SUBJ	4	ACM
117	TNS-ASP	2	OBL-COMPAR
117	ASPECT	2	UDF
103	TENSE	2	NEG-CONST
103	MOOD	2	COORD-FORM
98	_TOP	1	OBL-AG
75	$	1	XADJUNCT
72	OBJ	1	OBL-PERL
55	ADJUNCT	1	OBL-MOD
54	DEGREE	1	OBL-ADL
51	ATYPE		
36	PASSIVE		
28	NEG		
23	PTYPE		
20	POSS		
19	XCOMP-PRED		
14	_VOC		
13	CLAUSE-TYPE		

Figure 9: Values for the *cat* and *edge* attributes in POLFIE

million words) and by far the largest LFG treebank available in INESS. It was parsed with the eponymous grammar NorGram, a wide-coverage LFG grammar developed in the LOGON, TREPIL and INESS projects. Several versions of this grammar were constructed and used for parsebanking, including versions with c-structure pruning (Cahill et al. 2008). Some material was disambiguated manually with discriminants, but the bulk of the parsebank was disambiguated automatically through stochastic parseranking, based on the stored discriminants.

The collection NorGramBank (Dyvik et al. 2016) consists of a subset of the texts parsed in the NorGram collection. NorGramBank has more than 160 million words and consists of a variety of text types; while some newspaper texts were included, edited fiction and nonfiction texts were preferred because these have a higher language quality and fewer errors. Any error in a sentence, whether typographical, orthographical or grammatical, will result in a failure to find the intended analysis on parsing. Some NorGram texts were excluded from NorGramBank because the source texts had many OCR errors.

The text selection for the corpus was partially dependent on available resources. While published texts are valued sources for treebanks and other corpora, copyright restrictions must be taken into account. It is therefore paramount to clear permissions with rights holders before starting to work on texts. In the case of NorGram, several texts were obtained through the National Library of Norway. For some of these, copyright had expired. For newer texts, exceptional permission to use these with some restrictions was obtained from the government. Every corpus must be provided with metadata, including such information as provenance and conditions for use.

The Norwegian treebanks parsed with NorGram have proved useful for lexicography (see Section 4.4). Some NorGram treebanks have been specifically added for NAOB, a dictionary project by the Norwegian Academy for Language and Literature aimed at building a large dictionary for Norwegian Bokmål. In INESS, the collection called NAOB consists of 15 treebanks with a total of over 11 million sentences (161 million words).

The Norwegian example analyses shown in Figures 10, 11, 14, 15 and 18 are all from the NorGram treebanks.

3.5 Small treebanks for grammar development

Most of the small LFG treebanks in INESS are test suites used in various projects. GeoGram, HunGram and WolGram are collections of test suites used for the development of XLE grammars for Georgian (Meurer 2009), Hungarian (Laczkó et al. 2013, Laczkó 2014) and Wolof (Dione 2014, 2019), respectively. Some of these

test suites are parts of parallel treebanks (see Section 3.6). Other treebanks in these collections may only be available to their creators since they are work in progress and not at a stage where they may be useful to other researchers. Treebank developers decide whether they want to make their treebanks publicly available.

3.6 Parallel treebanks with LFG annotations

A parallel treebank is a collection of monolingual treebanks that are aligned with each other on the sentence level, and sometimes also on phrase and/or word levels. The most common type of parallel treebank involves one or more translations of a text that are aligned with the source text, but a parallel treebank can also have different annotations of the same text, for example a constituency annotation and a dependency annotation.

The user can select aligned parallel treebanks by choosing *Show only Parallel Treebanks* on the *Treebank Selection* page and selecting a collection from those that are then displayed in boldface. One of the treebanks to be examined is then chosen in the usual manner by clicking in the box next to the treebank name and subsequently clicking on the treebank name. From the *Sentence Overview* page, clicking on *Treebank Details* provides an overview of which other treebanks are aligned. Selecting one of those treebanks will start the display of parallel analyses for the two chosen languages.

The following subsections will present the XPAR Project (Section 3.6.1), the treebanks developed in the Parallel Grammar Project (Section 3.6.2), and other parallel treebanks containing LFG analyses (Section 3.6.3).

3.6.1 The XPAR Project

Language Diversity and Parallel Grammars (XPAR) was a pilot project which aimed to determine to what extent the development of parallel deep grammars for typologically diverse languages may support the automatic derivation of high-quality parallel treebanks for those languages (Dyvik et al. 2009). Principles for phrase alignment and methodology for the automatic alignment of c-structures from manually aligned f-structures were developed in the project.

A small parallel test suite of translationally equivalent Georgian and Norwegian sentences was used in developing the alignment tool. An example of aligned sentences is provided in (3), and their sentence-aligned analyses are shown in Figure 10.

(3) a. Georgian
gia-s uqvars eka.
Gia-DAT loves Eka.NOM
'Gia loves Eka.'

b. Norwegian
Jon elsker Maria.
Jon loves Maria
'Jon loves Maria.'

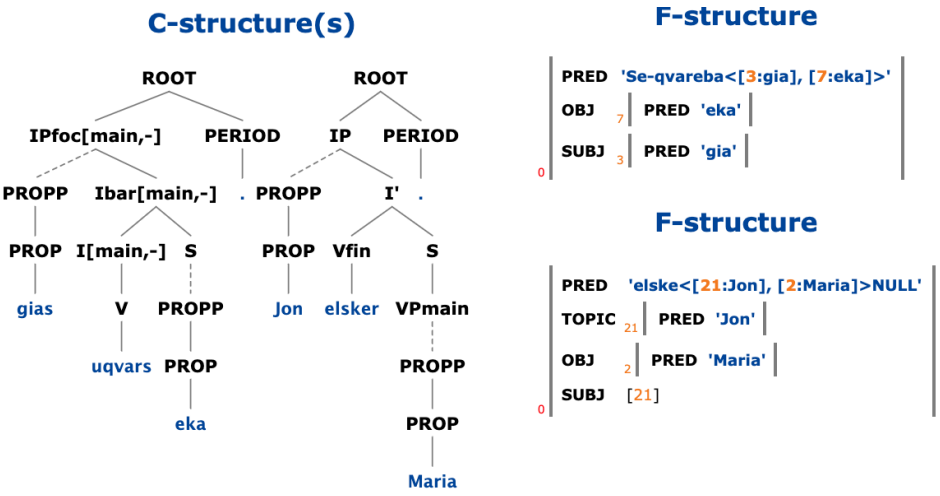


Figure 10: Sentence aligned c- and f-structures for the Georgian (http://hdl.handle.net/11495/D8B8-3970-851A-3@lfg51519) and Norwegian (http://hdl.handle.net/11495/D8B8-3970-851A-3@lfg60949) sentences in (3)

F-structures are manually aligned on the basis of translational correspondences at the level of predicate–argument structure. Subsidiary f-structures correspond if their predicates are in a translational relationship to one another. The alignment is done by dragging the index of one f-structure onto the corresponding index of the other f-structure. For instance, in Figure 10, the OBJ index 7 in the Georgian f-structure may be dragged onto the OBJ index 2 in the Norwegian one. This results in indices of the form $\boxed{n \rightarrow m}$, where n is the original index of that f-structure and m is the original index of the f-structure it is aligned with. Figure 11 shows the result of this manual alignment of f-structures, where the indices for the OBJ, SUBJ and main PRED have been aligned. Once the f-structures are aligned, the LFG Parsebanker automatically aligns the corresponding nodes

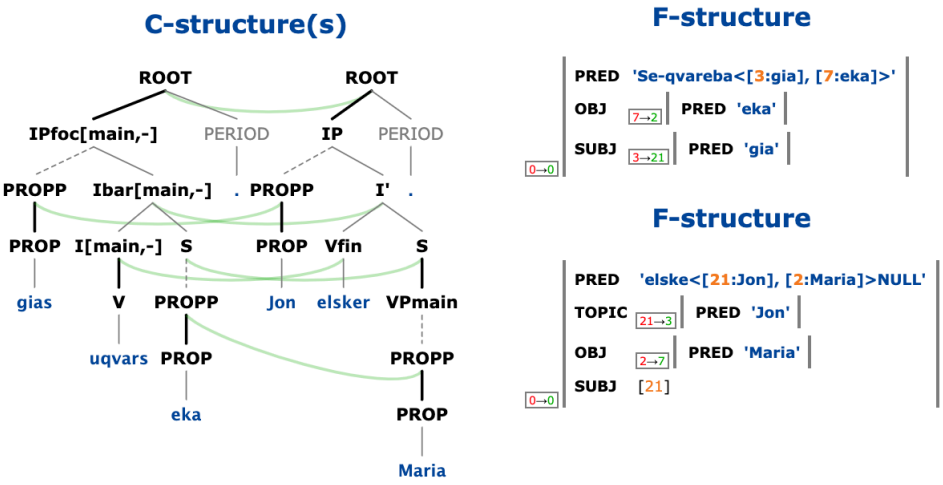


Figure 11: Word and phrase aligned c- and f-structures for the Georgian and Norwegian sentences in (3)

in the c-structures, shown by the curved green lines. We see, for example, that the OBJ alignment in the f-structures results in the alignment of the PROPP nodes dominating *Eka* and *Maria* in the c-structures.

3.6.2 The Parallel Grammar Project treebanks

The Parallel Grammar Project (ParGram) is an international cooperative effort to develop parallel LFG grammars implemented in XLE (Butt et al. 1999, 2002). Originally three languages were involved in the project: English, French and German; later, other languages joined, including Georgian, Hungarian, Indonesian, Japanese, Norwegian, Polish, Tamil, Turkish, Urdu and Wolof, among others. The main focus of the ParGram project was to develop and maintain linguistically motivated parallelism at the level of f-structure. Some of the ParGram participants have also been involved in the ParSem project, an effort to develop semantic structures based on the ParGram syntactic structures, with most of the ParSem systems using XLE's transfer system.

ParGram has created two parallel treebanks to support the aim of developing parallel LFG grammars. These treebanks consist of test suites encompassing various syntactic constructions. The English sentences were first agreed upon, and then translated into the other languages in the project. The first set of 50 sentences included such constructions as declaratives, interrogatives, imperatives, transitivity, passive, unaccusative, and subcategorized declaratives (Sulger et al.

2013). These sentences are included in the ParGram collection in INESS. Another set of sentences, concerned with tense, mode and aspect, constitutes the ParTMA collection. Figure 12 shows word and phrase aligned c- and f-structures for the English and German sentences in (4).

(4) a. What did the farmer see?

b. German
Was sah der Bauer?
what saw the farmer
'What did the farmer see?'

The f-structures for these sentences are practically identical, whereas the c-structures are quite different. This is both because the languages are different (English has *do*-support and German does not) and because the grammars for these languages have used quite different principles and techniques in writing the phrase structure rules. Still we see that most c-structure nodes are aligned. Since the XPAR principles align only translationally corresponding f-structures with PRED values, not all c-structure nodes can be aligned. The word *did* and the question marks only contribute features to the f-structure, not PRED values; these features are not shown here since the f-structures are displayed in PREDS only mode.

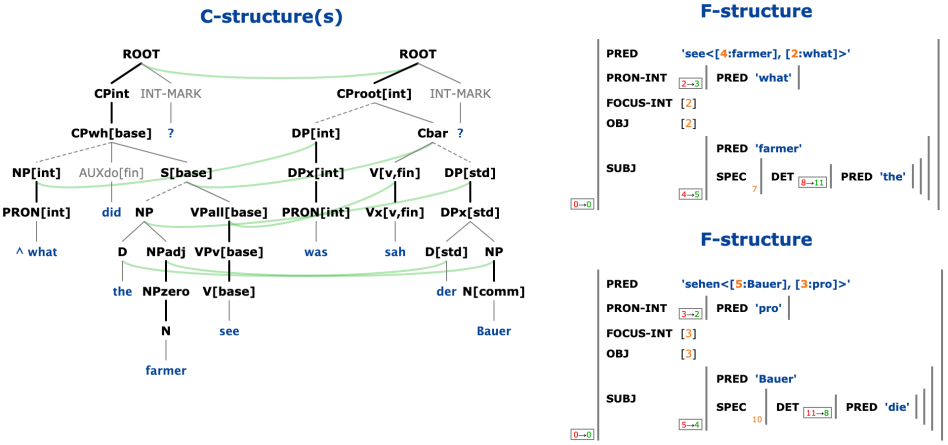


Figure 12: Word and phrase aligned c- and f-structures for the English (http://hdl.handle.net/11495/D8B8-3970-851A-3@lfg423651) and German (http://hdl.handle.net/11495/D8B8-3970-851A-3@lfg444239) sentences in (4)

3.6.3 Other parallel treebanks including LFG

Several projects have built parallel treebanks that include both LFG treebanks and treebanks of other types. Three such parallel treebanks are presented here.

The Sofie Parallel Treebank is a parallel corpus containing the first chapters of Jostein Gaarder's novel *Sofies verden* "Sophie's World". This text was chosen for treebanking because it is a well-written text that has been translated into a great number of languages. The Nordic Treebank Network developed treebanks based on these texts for Danish, Estonian, German, Icelandic and Swedish in the period 2001–2005. The META-NORD project,[14] which ran from 2011 to 2013, had as one of its goals to promote the accessibility of treebanks, including some that had not been maintained and were no longer accessible (Losnegaard et al. 2013). An English treebank, originally developed in the SMULTRON project,[15] and a Georgian treebank, developed at Uni Computing in Bergen, Norway, were added to the Sofie collection. Two treebanks for Norwegian were also developed, one an LFG treebank and the other a constituency treebank with syntactic and functional categories. Only the Georgian and one of the Norwegian treebanks have LFG annotation; the rest of the treebanks have various types of constituency annotation. In the initial version of the LFG Sofie treebank for Norwegian, 73% of sentences received analyses. An in-depth study of the sentences that received full parses that were not entirely correct showed that 29% lacked the correct analysis because of grammar problems, while lexical problems accounted for 71%, with missing multiword expressions in the lexicon being the most important of these. Subsequent grammar and lexicon updates resulted in correct analyses for more than 90% of these sentences (Losnegaard et al. 2012).

The META-NORD Acquis Parallel Treebank is a small parallel corpus of translations of a European Union directive.[16] The EU languages Danish, Estonian, Finnish, Latvian and Swedish, as well as the non-EU languages Norwegian and Icelandic, have treebanks in the collection. All language pairs are aligned at sentence level. The Norwegian treebank contains LFG analyses, while the other languages have consistency or dependency annotations.

The Norwegian Dependency Treebank was developed by the National Library of Norway (Solberg et al. 2014); it is made available in INESS as the treebanks named nob-ndt-dep (for Norwegian Bokmål) and nno-ndt-dep (for Norwegian

[14] http://www.meta-net.eu/projects/meta-nord/

[15] https://www.ling.su.se/english/nlp/corpora-and-resources/smultron/stockholm-multilingual-treebank-smultron-1.14047

[16] Directive 2002/74/EC, from the Acquis Communautaire (AC), the total body of European Union law applicable in the member states.

Nynorsk). The treebank has also been converted to the Universal Dependencies (UD) annotation scheme (Øvrelid & Hohle 2016), creating the treebanks nob-ud-2.5-dep and nno-ud-2.5-dep. The same texts were parsed with NorGram to obtain LFG analyses, resulting in the treebanks nob-ndt-lfg and nno-ndt-lfg. The original dependency annotations were created automatically, but the analyses were then manually checked and corrected, resulting in a gold standard treebank. The dependency treebanks contain analyses for all sentences, while the LFG treebank has coverage for about 90% of the sentences. The analyses for the sentences that are covered in the LFG treebank are, however, much more detailed than those in the dependency treebanks. See Section 4.5 for more on UD treebanks, including a comparison with LFG analyses.

4 Exploring and exploiting LFG treebanks

4.1 INESS Search

Prior to the INESS project, there was no search tool that could perform search in LFG f-structures. INESS Search (Meurer 2012, 2020, Rosén et al. 2017) is a search tool that was developed in order to fill this need. It is a reimplementation and extension of TIGERSearch (Lezius 2002), a search system designed for the TIGER treebank (Zinsmeister et al. 2002, Brants et al. 2004). INESS Search retains the full functionality of TIGERSearch for querying constituency and dependency treebanks while extending its functionality in order to query fully general directed graphs like LFG f-structures; in addition, it can be used for search in HPSG treebanks. INESS Search supports almost full first-order predicate logic, including negation and existential and universal quantification, with the exception of universal quantification over disjunctions.

INESS Search is fully integrated in the INESS infrastructure and is used via its Web interface. There is extensive documentation for INESS Search online, both a walkthrough that describes how to get started searching in INESS treebanks,[17] and thorough documentation of the query language itself.[18]

In addition to extending TIGERSearch, INESS Search has implemented simplifications to the syntax of search expressions for more clarity. Suppose you want to find examples of NPs with AP modifiers that have embedded PPs, such as the German NP in (5). In TIGERSearch you could write the search expression in (6), whereas (7) is an equivalent abbreviated expression in INESS Search.

[17] https://clarino.uib.no/iness/page?page-id=INESS_Search_Walkthrough
[18] https://clarino.uib.no/iness/page?page-id=INESS_Search

(5) German
die von Slumbewohnern unerlaubt gebauten Lehmhütten
the by slum.dwellers illegally built mud.huts
'the mud huts illegally built by slum dwellers'

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

(7) `NP > AP > PP`

The TIGERSearch expression in (6) may be read as follows: "There is a node with the category NP that dominates a node #*x* with the category AP; this same AP node #*x* dominates a node with the category PP." Each node has a variable, but it does not always need to be expressed; in (6), it is necessary to specify through the use of an explicit variable that it is the same AP that is dominated by the NP and that dominates the PP, otherwise the search results would return all sentences where there is at least one NP dominating an AP and at least one AP dominating a PP. In the abbreviated INESS Search expression (7), this chaining is inferred, so that an explicit mention of the variable is not necessary in this case. Furthermore, as also shown in Table 1, node labels may be used directly in the search expression, lexical and terminal nodes need only be enclosed in double quotes, and atomic f-structure values only in single quotes. One of the search results for the search expression in (7) from the TIGER treebank, the NP in (5), is shown in Figure 13; the node labels mentioned in the search expression are highlighted in red in the graph.

Table 1: Some examples of abbreviated syntax in INESS Search

Expression	Abbreviation	Explanation
`[cat="NP"]`	`NP`	node labels
`[word="book"]`	`"book"`	lexical nodes in dependency treebanks; terminal nodes in LFG and phrase-structure treebanks
`[atom="sg"]`	`'sg'`	atomic f-structure values in LFG treebanks
`[PP > #x:NP & #x > PP]`	`PP > NP > PP`	chaining of relations

4.2 Querying with INESS Search

Formulating well-targeted search expressions presupposes knowledge about the analyses in the treebank. One way of quickly gaining such knowledge is to use

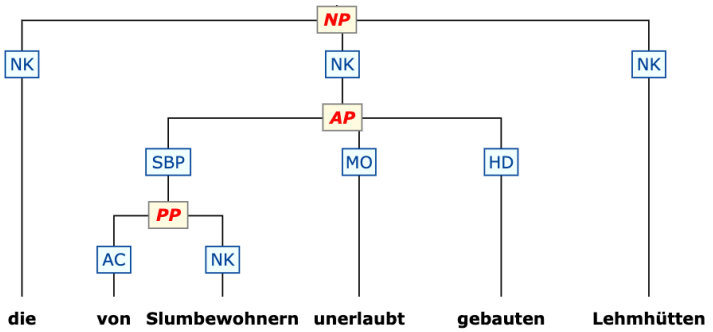


Figure 13: TIGER tree for (5) (http://hdl.handle.net/11495/D8B8-3970-851A-3@dep101299)

XLE-Web to parse sentences with the kind of grammatical phenomenon one is interested in and to study the analyses. Suppose that we want to search for passive sentences. The Norwegian passive sentence in (8) gets the analysis in Figure 14 when parsed in XLE-Web.

(8) Norwegian
Verden ble skapt av Gud.
world.DEF.SG was created by God
'The world was created by God.'

Examining the f-structure shows that the verb *skape* 'create' is the head of the XCOMP. It is a two-place predicate, with the PRED of the OBL-AG, *Gud* 'God', as its first argument, the agent. The XCOMP also has an attribute-value pair 'PASSIVE +'. A simple search expression for passives with agent phrases can thus be formulated using these f-structure characteristics, as shown in (9).

(9) `#x >PASSIVE #y:'+' & #x >OBL-AG`

This expression may be read: "There is an f-structure #*x* which has an attribute PASSIVE with the value '+' (bound to #*y*), and this same f-structure #*x* also has an attribute OBL-AG."

The negation operator in INESS Search allows users to restrict searches with respect to properties that sentences should *not* have. The search expression in (10), where the exclamation point is the negation operator, searches for passives *without* agent phrases. The sentence in (11) is one of those found by this expression; its c- and f-structures are shown in Figure 15. The f-structure nodes that are named with explicit variables in the search expression are marked in red in the

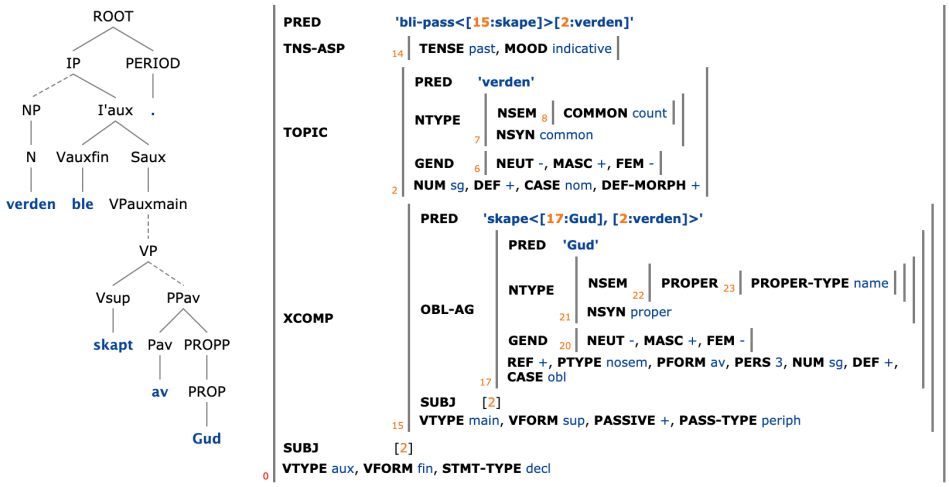


Figure 14: C- and f-structures from XLE-Web for the passive sentence in (8)

search result. In the f-structure we note that the XCOMP does not have an OBL-AG, and that the first argument of the main PRED is 'NULL'.

(10) `#x >PASSIVE #y:'+' & #x !>OBL-AG`

(11) Norwegian
Hvordan er verden skapt?
how is world.DEF.SG created
'How was the world created?'

4.3 An example-based introduction

For some researchers, INESS Search can be difficult to use, even with the simplifications that have been introduced. To assist users of NorGramBank in formulating search expressions, an example-based introduction to the search system has been written.[19] It is based on the Norwegian reference grammar *Norsk referansegrammatikk* (Faarlund et al. 1997) and the chapters and examples therein. Most researchers in Norwegian syntax will be familiar with the rather theory-neutral analyses in this book, and the goal is to provide them with LFG analyses of the

[19]This introduction, in Norwegian, is part of the INESS documentation: https://clarino.uib.no/iness/page?page-id=norgram-soek#innledning.

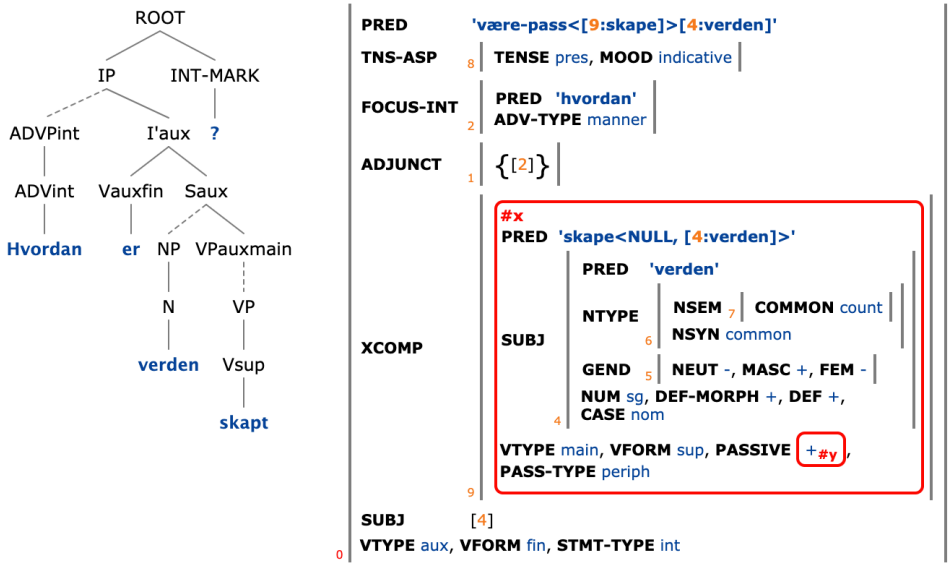


Figure 15: C- and f-structures for the passive sentence in (11) (http://hdl.handle.net/11495/D8B8-3970-851A-3@lfg6174124)

constructions that are of interest, including the page numbers in the book where the constructions are treated. For each construction, the example-based documentation provides an LFG analysis of one sentence together with a commentary explaining the analysis. A search expression that will find the construction is provided, along with both a paraphrase and a lengthier prose explanation of the expression. Finally a list of a few matching sentences is presented.

A construction type that is difficult to search for without a treebank is relative clauses without complementizers. It would not be straightforward to find these in corpora which are not syntactically annotated, so this is a good illustration of the added value of treebanks. The search expression for relatives without complementizers is given in (12).

(12)
```
#x >(ADJUNCT $) #f >TOPIC-REL #g
& #f >OBJ #g & #f >CLAUSE-TYPE 'rel'
& !(#f >COMP-FORM)
& !(#x >PRON-TYPE 'free')
```

This search expression may be read: "An f-structure #x has an attribute ADJUNCT with a value that includes an f-structure #f; furthermore, #f has an attribute TOPIC-REL with the value #g, and an attribute OBJ with the same value #g;

#*f* also has an attribute CLAUSE-TYPE with the value 'rel' and does not have an attribute COMP-FORM; the f-structure #*x* does not have an attribute PRON-TYPE with the value 'free' (the last specification ensures that free relatives will not be found)." An example sentence found by this expression in NorGramBank is given in (13), where the boldfaced relative clause *jeg så* lacks a complementizer.

(13) Norwegian
Alt **jeg så** var frontlykt-ene.
all I saw was headlight-DEF.PL
'All I saw was the headlights.'

4.4 Search with templates

A further simplification in INESS Search is the implementation of search templates, which abbreviate complete parameterized search expressions. For the Norwegian treebank NorGramBank, a number of such templates have been provided, primarily for the benefit of lexicographers.[20] Templates obviate the need for understanding an often complicated search expression, since users can choose one on the basis of a description of its intention, but they can examine the whole expression if desired. Templates are parameterized in the sense that the user can fill in values for one or more parameters, such as word or lemma forms, predicates, or grammatical features.

Suppose you want to find out how common nominal complement clauses with and without complementizers are after certain verbs. The template shown in Figure 16, named *AT-verbwithandwithout(@verb)*, may be used for this purpose. The user fills in the verb, in this case *fortelle* 'tell, relate', and clicks on *Run query*. The results of the search are presented in a table, sorted according to whether they include the complementizer or not. We see that the vast majority of occurrences of complement clauses with this verb, 21,465 (97.5%), do have complementizers.

This can be compared with the results for the verb *tro* 'think, be of the opinion', shown in Figure 17. For this verb the proportion of uses with the complementizer is only 33.8%. In this screenshot the user has clicked on the first row in the table, showing the number of occurrences for the verb without the complementizer (66,258). This brings up a list over all the sentences with this pattern. Here the user has clicked twice on *Next* in order to come to page 3; there are so many hits that the list consists of 3,313 pages. When the user mouses over a sentence, a simplified f-structure is displayed to the right of the list. Clicking on a sentence

[20]Documentation in Norwegian: https://clarino.uib.no/iness/documentation/INESS-Sketch-veiledning-2020.pdf

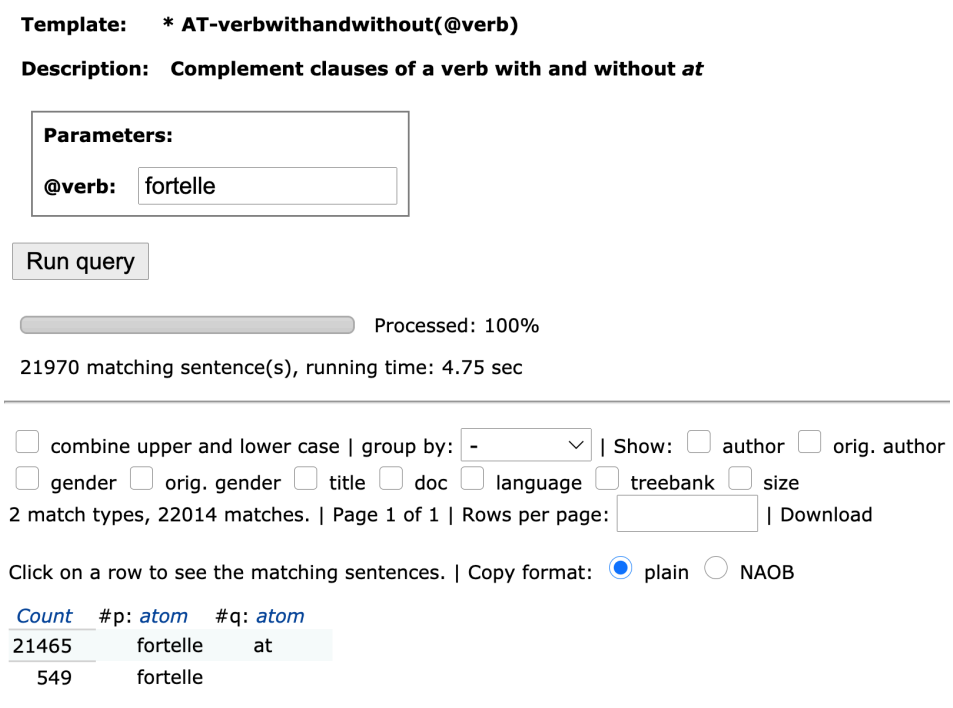
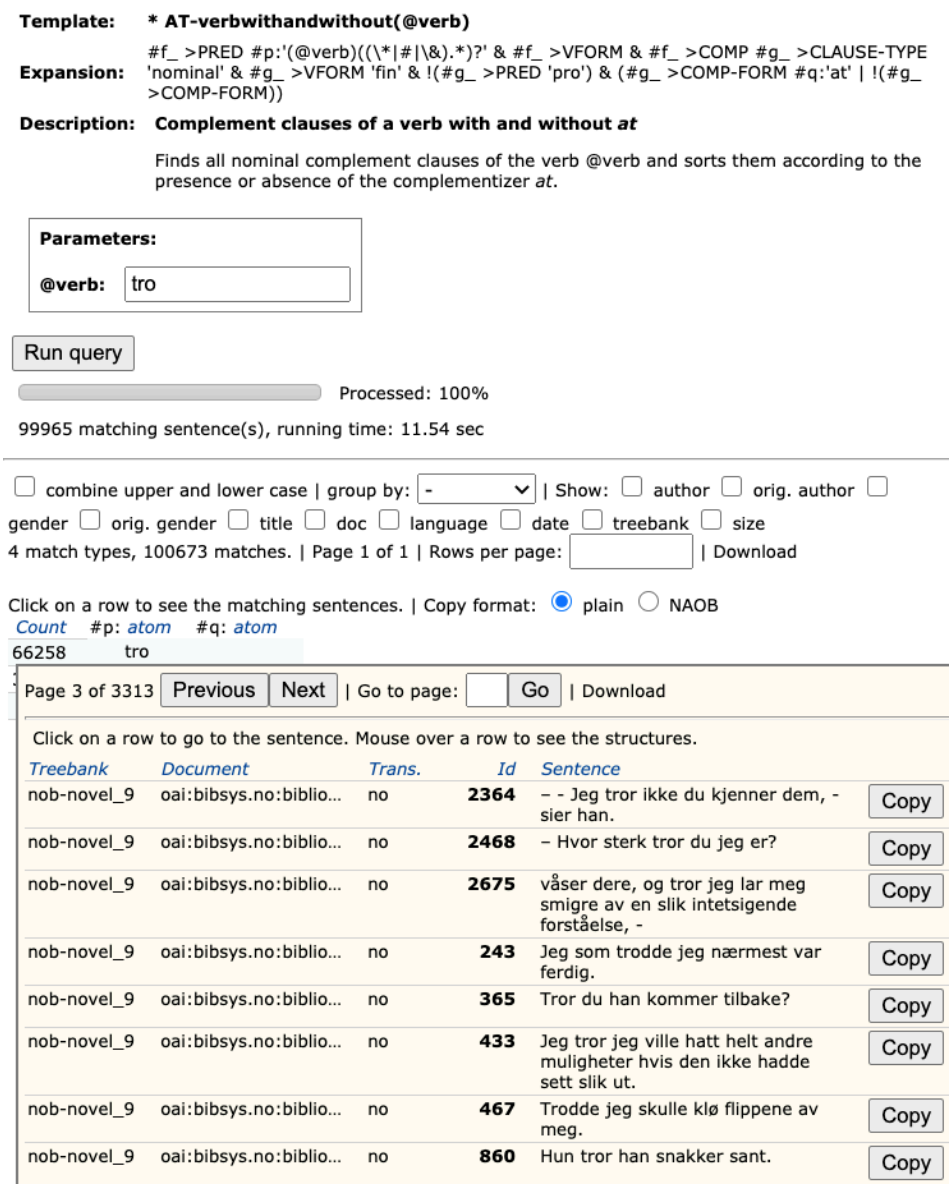


Figure 16: Template for nominal clause search with and without complementizer for the verb *fortelle* 'tell, relate'

brings the user to the *Sentence* page where the c-structure and the full f-structure are displayed. By default the quite complicated search expression which is used in this template is hidden, as in Figure 16. In Figure 17, the user has clicked on the template name, bringing up the expansion with the search expression. In this figure a more detailed prose description is also displayed, obtained by clicking on the boldfaced, more compact, part of the description.

Rauset et al. (2021) provide concrete examples of the use of template search in NorGramBank for various dictionary projects in Norway. The lexicographers use templates to examine both the usage and frequency of words. The most common valency frames for verbs, as well as the most common prepositions and/or particles that they occur with, are examined by using the template *V-argframes(@V)*; this template also provides evidence about whether the verbs occur reflexively. The templates *ADJ-attrib-or-nominal(@ADJ)* and *V-attr-or-pred-ptc(@V)* provide evidence of the nominal and adjectival use of participles, which is sometimes the basis for the creation of separate entries for derived adjectives.



Template: *** AT-verbwithandwithout(@verb)**

Expansion: #f_ >PRED #p:'(@verb)((*|#|\&).*)?' & #f_ >VFORM & #f_ >COMP #g_ >CLAUSE-TYPE 'nominal' & #g_ >VFORM 'fin' & !(#g_ >PRED 'pro') & (#g_ >COMP-FORM #q:'at' | !(#g_ >COMP-FORM))

Description: **Complement clauses of a verb with and without *at***

Finds all nominal complement clauses of the verb @verb and sorts them according to the presence or absence of the complementizer *at*.

Parameters:

@verb: tro

Run query

Processed: 100%

99965 matching sentence(s), running time: 11.54 sec

combine upper and lower case | group by: - | Show: author orig. author gender orig. gender title doc language date treebank size

4 match types, 100673 matches. | Page 1 of 1 | Rows per page: | Download

Click on a row to see the matching sentences. | Copy format: plain NAOB

Count	*#p: atom*	*#q: atom*
66258	tro	

Page 3 of 3313 Previous Next | Go to page: Go | Download

Click on a row to go to the sentence. Mouse over a row to see the structures.

Treebank	*Document*	*Trans.*	*Id*	*Sentence*	
nob-novel_9	oai:bibsys.no:biblio...	no	**2364**	– - Jeg tror ikke du kjenner dem, - sier han.	Copy
nob-novel_9	oai:bibsys.no:biblio...	no	**2468**	– Hvor sterk tror du jeg er?	Copy
nob-novel_9	oai:bibsys.no:biblio...	no	**2675**	våser dere, og tror jeg lar meg smigre av en slik intetsigende forståelse, -	Copy
nob-novel_9	oai:bibsys.no:biblio...	no	**243**	Jeg som trodde jeg nærmest var ferdig.	Copy
nob-novel_9	oai:bibsys.no:biblio...	no	**365**	Tror du han kommer tilbake?	Copy
nob-novel_9	oai:bibsys.no:biblio...	no	**433**	Jeg tror jeg ville hatt helt andre muligheter hvis den ikke hadde sett slik ut.	Copy
nob-novel_9	oai:bibsys.no:biblio...	no	**467**	Trodde jeg skulle klø flippene av meg.	Copy
nob-novel_9	oai:bibsys.no:biblio...	no	**860**	Hun tror han snakker sant.	Copy

Figure 17: Template for nominal clause search with and without complementizer for the verb *tro* 'think, be of the opinion'

Targeted queries that provide evidence for colligations are useful when treating high-frequency words with many senses. The template *N-argofverbs(@N)* provides a list sorted by frequency of the verbs that occur with a certain noun as their first or second argument. Such results help lexicographers determine whether the sense distinctions made in older versions of the dictionaries are still reasonable, or whether there should be changes made by adding or removing distinctions, or for instance by promoting a sense that is now more common than previously.

An example of a word which was missing a sense is the reflexive verb *utmerke seg* 'distinguish oneself', which was defined as having only a positive connotation. The lexicographers, however, did not believe this to be accurate. The template *V-prepobj(@V,@P)* was used to examine which words occur as objects of the prepositions *med* 'with' and *ved* 'by'. The search results showed several occurrences of the noun *mangel* 'lack' as the object of *ved*; one of these examples is given in (14). This and similar searches provided empirical support for the establishment of a new subsense of the verb with a negative connotation.

(14) Norwegian (http://hdl.handle.net/11495/D8B8-3970-851A-3@lfg14979442)
Han vil ... utmerke seg med mangel på konsistens i sine
he will distinguish REFL with lack of consistency in his
handlingsvalg ...
action.choice
'He will ... distinguish himself with lack of consistency in his choice of actions ... '

4.5 Comparison of search in LFG and dependency treebanks

Dependency treebanks are the most widely used type of treebanks, notably via the Universal Dependencies (UD) initiative.[21] The UD treebanks are grounded in dependency grammar, which assigns dependency relations between words, and does not analyze phrases and constituency relations (Tesnière 1959). An important early dependency treebank was the Prague Dependency Treebank (Hajič et al. 2001). Among the treebanks provided by INESS, dependency treebanks are the most numerous (250), with the UD treebanks accounting for most of these (200). The latest version in INESS at the time of writing is 2.8. INESS also keeps earlier versions, making it possible to track progress between versions.

The LFG and UD analyses of the sentence in (15) are shown in Figures 18 and 19. For both treebanks, information about lemma, part of speech and morphological

[21] https://universaldependencies.org

features may be displayed (by clicking on the word for the dependency treebank, or by clicking on the preterminal node for the LFG treebank). The c-structure in Figure 18 shows the hierarchical phrase structure of the sentence, labeled with a rich inventory of syntactic categories. The corresponding f-structure encodes syntactic functions, grammatical features, and predicate–argument relations, as represented in the semantic forms of the verbs. The dependency structure in Figure 19 is shallower and less detailed than the LFG structure. Dependencies between words are shown by labeled arrows that go from a word to its dependents.

(15) Norwegian
Han hadde aldri vært lykkeligere.
he had never been happier
'He had never been happier.'

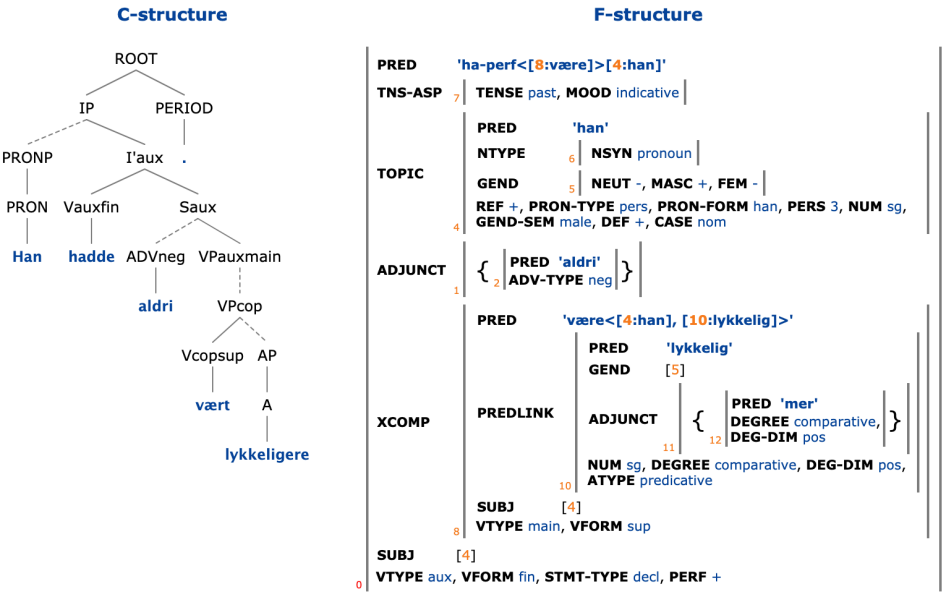


Figure 18: LFG analysis of the sentence in (15) (http://hdl.handle.net/11495/D8B8-3970-851A-3@lfg4292653)

The deeper analysis in an LFG treebank improves the search possibilities as compared with a dependency treebank. Rosén et al. (2020) compares search in the UD version of the Norwegian Dependency Treebank with the same texts in NorGramBank. The example given there is searching for the first argument of verbs. This may be done straightforwardly in an LFG treebank, but it is much

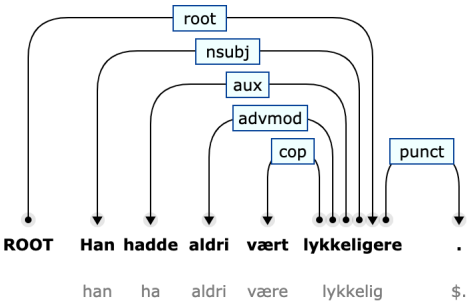


Figure 19: UD analysis of the sentence in (15) (http://hdl.handle.net/11495/D8B8-3970-851A-3@dep8965528)

more difficult in a dependency treebank since predicate–argument structure is not encoded there. The first argument of a verb can be the subject of an active verb or of a predicative present participle, the agent phrase of a passive verb, or the head of an attributive present participle. And since the UD guidelines allow for several ways of annotating some of these possibilities, creating a search expression to capture them is extremely complicated. For more detail on this comparison, see Rosén et al. (2020).

5 Conversion between LFG treebanks and other treebanks

Besides pure parsebanking with a grammar, other approaches have been used to construct treebanks by converting between formalisms or by enriching treebanks with additional information. The Universal Dependencies initiative is in some ways similar to ParGram in that both approaches aim at assigning common annotations to comparable items and structures across languages.

Since dependency relations may be labeled as grammatical functions such as subject and object, dependency structures have a resemblance to f-structures in LFG. The PARC 700 Dependency Bank is a treebank in dependency format based on the English LFG grammar developed at PARC (King et al. 2003). The corpus was created only to make a dependency bank. LFG analyses were transformed to dependency graphs, but no LFG treebank per se was created.

The TIGER corpus, mentioned in Section 3.2, utilized the large-scale German LFG grammar of the ParGram project for the semiautomatic creation of TIGER treebank annotations. The grammar was used for full parsing, followed by semiautomatic disambiguation and automatic transfer into the treebank format (Zinsmeister et al. 2002). The hybrid representation structure of TIGER, combining

constituent analysis and functional dependencies, benefited from information in the c-structures and f-structures provided by the LFG grammar.

Conversely, an LFG treebank may be created by enriching phrase-structure oriented treebank resources with functional structures, as suggested by Frank et al. (2003) and Cahill (2004). For more on grammar induction, see Cahill & Way 2023 [this volume].

Forst (2003) describes a method for converting the TIGER treebank to a test-suite for the German LFG ParGram grammar. The conversion utilizes the machine translation transfer system in XLE.

Recently, detailed algorithms for the conversion from LFG analyses to dependency structures were proposed by Meurer (2017) and Przepiórkowski & Patejuk (2020). While the latter follow the more standard assumption that f-structures provide a good basis for developing dependency trees, the former takes c-structures as the starting point, but combines this with information from f-structures.

6 Conclusion

This chapter has provided an introduction to LFG treebanks, illustrated throughout with the tools and visualizations of the INESS treebanking infrastructure. The process of developing an LFG grammar in tandem with a treebank through incremental parsebanking has been described. Both large and small LFG parsebanks for a number of languages have been presented. Several different methods for searching LFG treebanks with INESS Search have been explained: users can write search expressions themselves with the aid of XLE-Web and the INESS Search documentation; they can find search expressions for the phenomena they are interested in by consulting the example-based search documentation; and they can use search templates that only require filling in one or more search items. LFG treebanks have been compared with other treebanks, and it has been shown that the more detailed and sophisticated annotation in LFG treebanks provides richer opportunities for research than simpler annotations.

While INESS has already been developed over more than a decade, the system, and especially its interface, will continue to evolve. Consequently, future interactions may be slightly different from the interactions and screen displays shown in this chapter.

Although LFG treebanks are certainly valuable resources for research and development, building an LFG treebank is a time-consuming and expensive undertaking, especially for a language for which no large-coverage LFG grammar and lexicon yet exist. However, the task is made somewhat easier with the help of

the LFG Parsebanker as described above, and INESS is open to making more treebanks accessible for research and development.

Acknowledgments

I thank Koenraad De Smedt, Stefanie Dipper, Helge Dyvik, Jonas Kuhn, Paul Meurer, Agnieszka Patejuk, Heike Zinsmeister and three anonymous reviewers for helpful comments and suggestions. Thanks are also due to Mary Dalrymple for all of her help and her boundless patience.

References

Brants, Sabine, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith & Hans Uszkoreit. 2004. TIGER: Linguistic interpretation of a German corpus. *Research on Language and Computation* 2(4). 597–620. DOI: 10.1007/s11168-004-7431-3.

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius & George Smith. 2002. The TIGER treebank. In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories*, 24–41.

Brants, Thorsten & Oliver Plaehn. 2000. Interactive corpus annotation. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC'00)*. http://www.lrec-conf.org/proceedings/lrec2000/pdf/334.pdf.

Butt, Miriam, Stefanie Dipper, Anette Frank & Tracy Holloway King. 1999. Writing large-scale parallel grammars for English, French and German. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '99 conference*. Stanford: CSLI Publications.

Butt, Miriam, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi & Christian Rohrer. 2002. The Parallel Grammar Project. In John Carroll, Nelleke Oostdijk & Richard Sutcliffe (eds.), *COLING-GEE '02: Proceedings of the 2002 workshop on Grammar Engineering and Evaluation*, 1–7. Taipei: Association for Computational Linguistics. DOI: 10.3115/1118783.1118786.

Cahill, Aoife. 2004. *Parsing with automatically acquired, wide-coverage, robust, probabilistic LFG approximations*. Dublin: School of Computing, Dublin City University. (Doctoral dissertation).

Cahill, Aoife, John T. III Maxwell, Paul Meurer, Christian Rohrer & Victoria Rosén. 2008. Speeding up LFG parsing using c-structure pruning. In *COLING 2008: Proceedings of the workshop on Grammar Engineering Across Frameworks*, 33–40. Manchester. DOI: 10.3115/1611546.1611551.

Cahill, Aoife & Andy Way. 2023. Treebank-driven parsing, translation and grammar induction using LFG. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*, 1125–1167. Berlin: Language Science Press. DOI: 10.5281/zenodo.10185989.

Crouch, Richard, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, John T. III Maxwell & Paula S. Newman. 2011. *XLE Documentation*. Xerox Palo Alto Research Center. Palo Alto, CA. https://ling.sprachwiss.uni-konstanz.de/pages/xle/doc/xle_toc.html.

Dione, Cheikh M. Bamba. 2014. LFG parse disambiguation for Wolof. *Journal of Language Modelling* 2(1). 105–165. DOI: 10.15398/jlm.v2i1.81.

Dione, Cheikh M. Bamba. 2019. Clause structure, pro-drop and control in Wolof: An LFG/XLE perspective. *Nordic Journal of African Studies* 28(3). 1–26.

Dyvik, Helge, Paul Meurer, Victoria Rosén & Koenraad De Smedt. 2009. Linguistically motivated parallel parsebanks. In Marco Passarotti, Adam Przepiórkowski, Sabine Raynaud & Frank Van Eynde (eds.), *Proceedings of the 8th International Workshop on Treebanks and Linguistic Theories*, 71–82. Milan: EDUCatt.

Dyvik, Helge, Paul Meurer, Victoria Rosén, Koenraad De Smedt, Petter Haugereid, Gyri Smørdal Losnegaard, Gunn Inger Lyse & Martha Thunes. 2016. NorGramBank: A 'deep' treebank for Norwegian. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, 3555–3562. Portorož. http://www.lrec-conf.org/proceedings/lrec2016/summaries/943.html.

Faarlund, Jan Terje, Svein Lie & Kjell Ivar Vannebo. 1997. *Norsk referansegrammatikk*. Oslo: Universitetsforlaget.

Forst, Martin. 2003. Treebank conversion – Establishing a testsuite for a broad-coverage LFG from the TIGER treebank. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03) at EACL 2003*, 205–216. Association for Computational Linguistics. https://www.aclweb.org/anthology/W03-2404.

Forst, Martin & Tracy Holloway King. 2023. Computational implementations and applications. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*, 1083–1123. Berlin: Language Science Press. DOI: 10.5281/zenodo.10185986.

Frank, Anette, Louisa Sadler, Josef van Genabith & Andy Way. 2003. From treebank resources to LFG f-structures. In Anne Abeillé (ed.), *Treebanks: Building and using parsed corpora*, 367–389. Dordrecht: Kluwer Academic Publishers. DOI: 10.1007/978-94-010-0201-1_21.

Hajič, Jan, Barbora Vidová-Hladká & Petr Pajas. 2001. The Prague Dependency Treebank: Annotation structure and support. In *Proceedings of the IRCS Workshop on Linguistic Databases*, 105–114. Philadelphia.

King, Tracy Holloway, Richard Crouch, Stefan Riezler, Mary Dalrymple & Ronald M. Kaplan. 2003. The PARC 700 Dependency Bank. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora, held at the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, 1–8. Budapest: Association for Computational Linguistics.

Laczkó, Tibor. 2014. Essentials of an LFG analysis of Hungarian finite sentences. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '14 conference*, 325–345. Stanford: CSLI Publications.

Laczkó, Tibor, György Rákosi, Ágoston Tóth & Gábor Csernyi. 2013. Nyelvtanfejlesztés, implementálás és korpuszépítés: A HunGram 2.0 és a HG-1 Treebank legfontosabb jellemzői [Grammar development, implementation and corpus construction: Key features of HunGram 2.0 and the HG-1 Treebank]. In Attila Tanács & Veronika Vincze (eds.), *IX. Magyar Számítógépes Nyelvészeti Konferencia konferenciakötete*, 85–96. Szeged: JATEPress.

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

Losnegaard, Gyri Smørdal, Gunn Inger Lyse, Anje Müller Gjesdal, Koenraad De Smedt, Paul Meurer & Victoria Rosén. 2013. Linking Northern European infrastructures for improving the accessibility and documentation of complex resources. In Koenraad De Smedt, Lars Borin, Krister Lindén, Bente Maegaard, Eiríkur Rögnvaldsson & Kadri Vider (eds.), *Proceedings of the workshop on Nordic Language Research Infrastructure at NODALIDA 2013, May 22–24, 2013, Oslo, Norway. NEALT Proceedings Series 20* (Linköping Electronic Conference Proceedings 89), 44–59. Linköping University Electronic Press.

Losnegaard, Gyri Smørdal, Gunn Inger Lyse, Martha Thunes, Victoria Rosén, Koenraad De Smedt, Helge Dyvik & Paul Meurer. 2012. What we have learned from Sofie: Extending lexical and grammatical coverage in an LFG parsebank. In Jan Hajič, Koenraad De Smedt, Marko Tadić & António Branco (eds.), *Proceedings of the META-RESEARCH Workshop on Advanced Treebanking at LREC'12*, 69–76. Istanbul. http://www.lrec-conf.org/proceedings/lrec2012/workshops/12.LREC%202012%20Advanced%20Treebanking%20Proceedings.pdf.

Maxwell, John T. III & Ronald M. Kaplan. 1993. The interface between phrasal and functional constraints. *Computational Linguistics* 19. 571–590.

Meurer, Paul. 2009. A computational grammar for Georgian. In Peter Bosch, David Gabelaia & Jérôme Lang (eds.), *Logic, language, and computation: 7th International Tbilisi Symposium on Logic, Language, and Computation, TbiLLC 2007, Revised selected papers* (Springer Lecture Notes in Artificial Intelligence 5422), 1–15. Berlin: Springer. DOI: 10.1007/978-3-642-00665-4_1.

Meurer, Paul. 2012. INESS-Search: A search system for LFG (and other) treebanks. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '12 conference*, 404–421. Stanford: CSLI Publications.

Meurer, Paul. 2017. From LFG structures to dependency relations. *Bergen Language and Linguistics Studies* 8. 183–201. DOI: 10.15845/bells.v8i1.1341.

Meurer, Paul. 2020. Designing efficient algorithms for querying large corpora. *Oslo Studies in Language* 11(2). 283–302. DOI: 10.5617/osla.8504.

Meurer, Paul, Helge Dyvik, Victoria Rosén, Koenraad De Smedt, Gunn Inger Lyse, Gyri Smørdal Losnegaard & Martha Thunes. 2013. The INESS treebanking infrastructure. In Stephan Oepen, Kristin Hagen & Janne Bondi Johannessen (eds.), *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)* (Linköping Electronic Conference Proceedings), 453–458. Linköping University Electronic Press. http://www.ep.liu.se/ecp/085/043/ecp1385043.pdf.

Meurer, Paul, Victoria Rosén & Koenraad De Smedt. 2020. Interactive visualizations in INESS. In Miriam Butt, Annette Hautli-Janisz & Verena Lyding (eds.), *LingVis: Visual analytics for linguistics*, 55–85. Stanford: CSLI Publications / University of Chicago Press.

Øvrelid, Lilja & Petter Hohle. 2016. Universal dependencies for Norwegian. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16)*, 1579–1585. Portorož: European Language Resources Association (ELRA).

Patejuk, Agnieszka & Adam Przepiórkowski. 2012. Towards an LFG parser for Polish: An exercise in parasitic grammar development. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, 3849–3852. European Language Resources Association (ELRA).

Patejuk, Agnieszka & Adam Przepiórkowski. 2014. Synergistic development of grammatical resources: A valence dictionary, an LFG grammar, and an LFG structure bank for Polish. In *Proceedings of the 13th International Workshop on*

Treebanks and Linguistic Theories (TLT13), 113–126. Department of Linguistics (SfS), University of Tübingen.

Pollard, Carl & Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press & CSLI Publications.

Przepiórkowski, Adam & Agnieszka Patejuk. 2020. From Lexical Functional Grammar to Enhanced Universal Dependencies: The UD-LFG treebank of Polish. *Language Resources and Evaluation* 54. 185–221. DOI: 10.1007/s10579-018-9433-z.

Rauset, Margunn, Gyri Smørdal Losnegaard, Helge Dyvik, Paul Meurer, Rune Kyrkjebø & Koenraad De Smedt. 2021. Words, words! Resources and tools for lexicography at the CLARINO Bergen Centre. Unpublished manuscript.

Rosén, Victoria & Koenraad De Smedt. 2022. Managing treebank data with the infrastructure for the exploration of syntax and semantics (INESS). In Andrea L. Berez-Kroeker, Bradley McDonnell, Eve Koller & Lauren B. Collister (eds.), *MIT open handbook of linguistic data management* (Open Handbooks in Linguistics), 499–512. Cambridge, MA: The MIT Press.

Rosén, Victoria, Koenraad De Smedt & Paul Meurer. 2006. Towards a toolkit linking treebanking to grammar development. In *Proceedings of the 5th Workshop on Treebanks and Linguistic Theories*, 55–66.

Rosén, Victoria, Koenraad De Smedt, Paul Meurer & Helge Dyvik. 2012. An open infrastructure for advanced treebanking. In Jan Hajič, Koenraad De Smedt, Marko Tadić & António Branco (eds.), *Proceedings of the META-RESEARCH workshop on Advanced Treebanking at LREC'12*, 22–29. Istanbul: European Language Resources Association (ELRA).

Rosén, Victoria, Helge Dyvik, Paul Meurer & Koenraad De Smedt. 2020. Creating and exploring LFG treebanks. In Miriam Butt & Ida Toivonen (eds.), *Proceedings of the LFG '20 conference*, 328–348. Stanford: CSLI Publications.

Rosén, Victoria, Helge J. Jakhelln Dyvik, Paul Meurer & Koenraad De Smedt. 2017. Exploring treebanks with INESS search. *Proceedings of the 21st Nordic Conference on Computational Linguistics (NoDaLiDa)*. Linköping Electronic Conference Proceedings 29(131). 326–329. http://www.ep.liu.se/ecp/131/048/ecp17131048.pdf.

Rosén, Victoria, Paul Meurer & Koenraad De Smedt. 2005. Constructing a parsed corpus with a large LFG grammar. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '05 conference*, 371–387. Stanford: CSLI Publications.

Rosén, Victoria, Paul Meurer & Koenraad De Smedt. 2007. Designing and implementing discriminants for LFG grammars. In Miriam Butt & Tracy Holloway

King (eds.), *Proceedings of the LFG '07 conference*, 397–417. Stanford: CSLI Publications.

Rosén, Victoria, Paul Meurer & 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 & Koenraad De Smedt (eds.), *Proceedings of the 7th International Workshop on Treebanks and Linguistic Theories (TLT7)*, 127–133. Utrecht: LOT.

Solberg, Per Erik, Arne Skjærholt, Lilja Øvrelid, Kristin Hagen & Janne Bondi Johannessen. 2014. The Norwegian Dependency Treebank. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*, 789–795. European Language Resources Association (ELRA).

Sulger, Sebastian, Miriam Butt, Tracy Holloway King, Paul Meurer, Tibor Laczkó, György Rákosi, Cheikh M. Bamba Dione, Helge Dyvik, Victoria Rosén, Koenraad De Smedt, Agnieszka Patejuk, Özlem Çetinoglu, I Wayan Arka & Meladel Mistica. 2013. ParGramBank: The ParGram parallel treebank. In *Proceedings of the 51st annual meeting of the Association for Computational Linguistics*, vol. 1, 550–560. Sofia: Association for Computational Linguistics. https://www.aclweb.org/anthology/P13-1054.pdf.

Tesnière, Lucien. 1959. *Éléments de syntaxe structurale*. Paris: Klincksieck.

Zinsmeister, Heike, Jonas Kuhn & Stefanie Dipper. 2002. TIGER transfer — Utilizing LFG parses for treebank annotation. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '02 conference*, 427–447. Stanford: CSLI Publications.