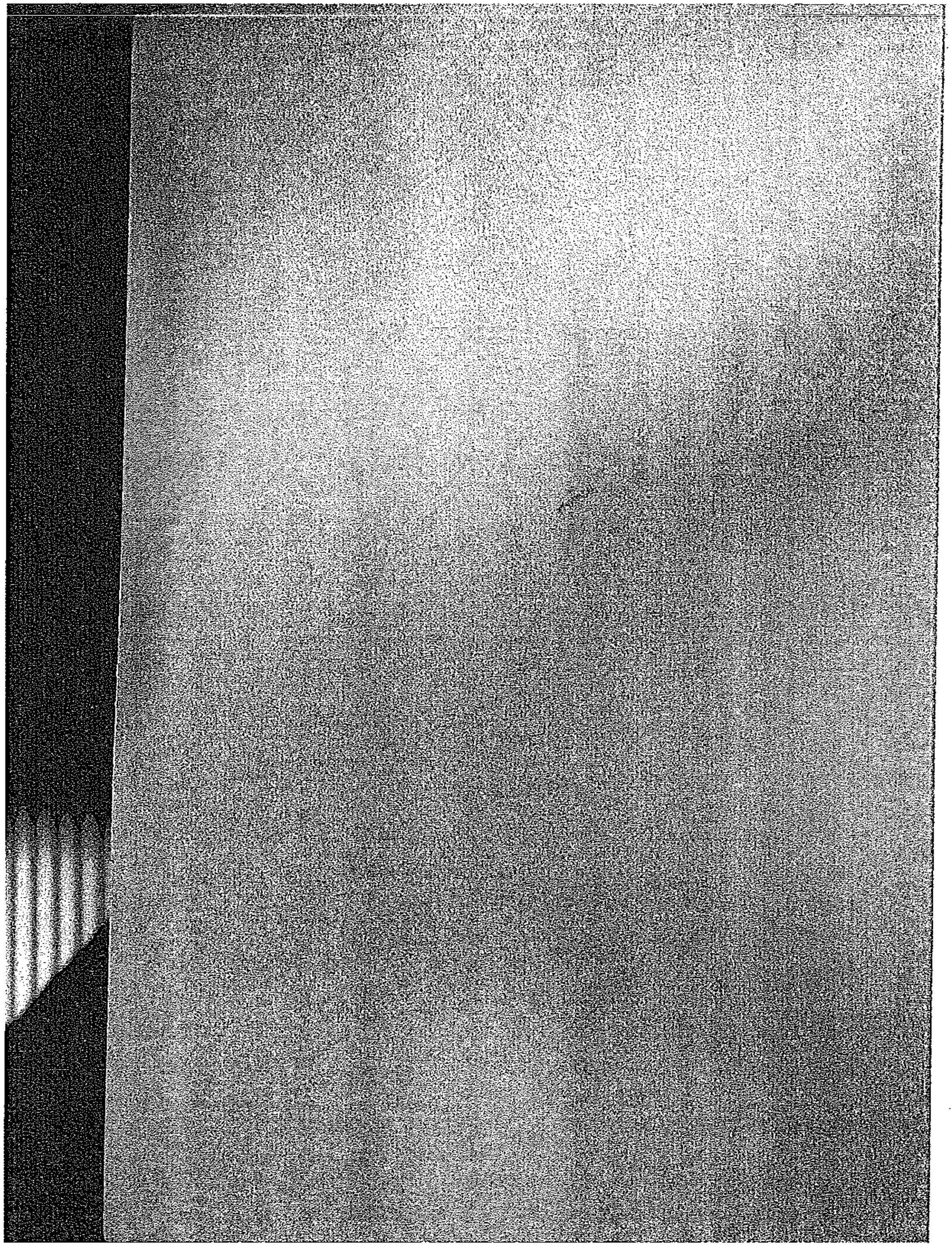


Janne Bondi Johannessen

Proceedings of the

Third Workshop on Treebanks and Linguistic Theories (TLT 2004)

Tübingen, December 10–11, 2004

organized by the

Collaborative Research Centre 441
"Linguistic Data Structures"
University of Tübingen, Germany

and the

Nordic Treebank Network

Editors

Sandra Kübler
Joakim Nivre
Erhard Hinrichs
Holger Wunsch

[8] Hoffmann, Walter/Wetter, Friedrich (1987): Bibliographie frühneuhochdeutscher Quellen: ein kommentiertes Verzeichnis von Texten des 14. -17. Jahrhunderts. 2. überarbeitete Auflage. Frankfurt/M.: Peter Lang.

[9] Lötscher, Andreas (2000): Verbendstellung im Hauptsatz in der deutschen Prosa des 15. und 16. Jahrhunderts. Sprachwissenschaft 25, 153-191.

[10] Schröder, Werner (1985): Auxiliar-Ellipsen bei Geiler von Kayserberg und bei Luther. Wiesbaden/Stuttgart: Steiner.

[11] Solms, Hans-Joachim/Wegera, Klaus-Peter (1998): Das Bonner Frühneuhochdeutschkorpus. Rückblick und Perspektiven. In: Rolf Bergmann (Hg.): Probleme der Textauswahl für einen elektronischen Thesaurus. Beiträge zum ersten Göttinger Arbeitsgespräch zur historischen deutschen Wortforschung am 1. und 2. November 1996. Stuttgart: Hirzel, 22-39.

[12] Stolt, Birgit (1990): Redeglieder, Informationseinheiten: Cola und Commata in Luthers Syntax. In: Anne Betten (Hg.): Neuere Forschungen zur historischen Syntax des Deutschen. Tübingen: Niemeyer, 379-392.

SearchTree - A user-friendly treebank search interface

Lars Nygaard and Janne Bondi Johannessen

University of Oslo
The Text Laboratory
http://www.hf.uio.no/tekstlab/
{larsnyg, jannebj}@ilf.uio.no

1 Introduction

Treebanks constitute a valuable resource for linguists, but their usefulness is often reduced by hard-to-use search interfaces, often requiring the user to learn the detailed knowledge of query languages or regular expressions as well as of tag sets, often with non-intuitive tag names and abbreviations. Writing complex queries becomes a slow and error-prone process. In addition, the user will often have to learn several query languages, with smaller and larger differences, adding to the confusion.

We think that user-friendliness is as important for treebank use as it is for the use of text corpora generally. In this paper we describe SearchTree, a web-based interface for queries in treebanks. SearchTree is not tied to any particular treebank, although its main motivation comes from the need for a proper search interface for the Sofie Treebank – a parallel treebank of mainly North European languages (Danish, Dutch, English, Estonian, Faroese, Finnish, German, Icelandic, Norwegian, Swedish).[1]

In the following, we will provide a description of SearchTree, and exemplify with monolingual searches in the Penn Treebank, and with parallel searches in the Sofie Treebank. We will then briefly perform a a comparison with other treebank search interfaces.

2 The advantages of SearchTree

SearchTree is implemented in HTML, JavaScript and Perl. As a search engine it uses TGrep2 (Rohde 2004), a query engine for linguistically annotated trees. TGrep2 has good functionality for many kinds of tree search, but cannot deal

[1] The Sofie Treebank, which is still under development, is the result of joint work of the members of the Nordic Treebank Network, http://w3.msi.vxu.se/~nivre/research/nt.html.

with crossing branches or secondary edges. An additional underlying query system will be provided in the near future to handle these.

There are several advantages in the SearchTree system.

- SearchTree is publically available; it is open source (see the reference list).
- It is accessible to the user via a web browser; no installation is necessary.
- SearchTree provides all tags and categories that are used in the searchable treebank(s); the user need not learn the tagsets before starting the queries.
- SearchTree is completely graphical in an intuitive interface.
- The results are presented in a user-friendly way, showing the query subtree, plus the whole sentence as text, and with the option of seeing the whole tree.
- SearchTree is not tied to any specific corpus or formalism.

Most users will be linguists with little wish to learn the syntax and terminology of a complex search language like TGrep2. We think that the user is better served by clicking in boxes than having to formulate complex queries in a complex query language. But other users will also be pleased not having to face the danger of putting a parenthesis in the wrong place etc. Furthermore, users should not need to know by heart all the names of parts of speech and categories used. Such expressions should be given as lists.

3 The SearchTree query interface

In this section, we will present the SearchTree query interface. The basic features of the web interface for monolingual searches are illustrated below:

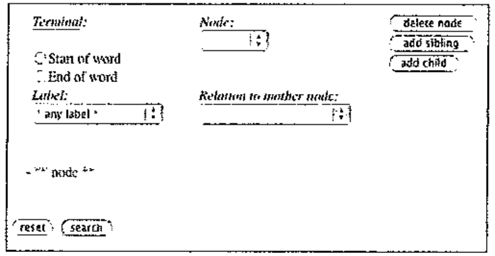


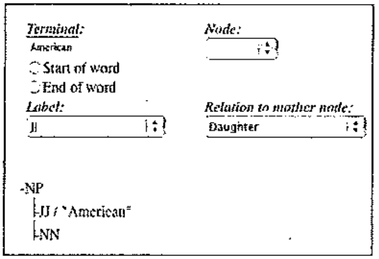
Figure 1

The user starts by clicking on the red, highlighted string node, activating this as the first node. The next step is to choose a label for this node from the pull-down menu above it. A new node can now be added by clicking on one of the boxes on the right-hand side, which will insert a new node in the required place (i.e., as sibling or daughter). Again a label for this node must be chosen. Further specifications can be picked from the pull-down menu for relations between nodes, or for writing a terminal in the box on the upper left, or for specifying the "modality" of the node; optional, negated or obligatory. The user can only activate one node at the time. Any option that is being chosen will apply to that particular activated node. We will illustrate this.

Let us say that we are interested in NPs that contain at least the adjective *American* and an optional, common noun. This would give the following TGrep2 query:

(1) (NP < (JJ < /^American$/)?< (NN))

In order to be able to write this query, the user would ordinarily have to be able to know the syntactic tags, and the syntax and inventory of the TGrep2 search language. In SearchTree, the query is formulated instead as in figure 2, with results given in figure 3:


Terminal: American

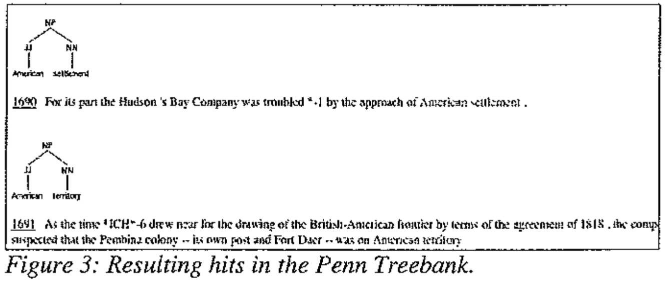
○ Start of word

○ End of word

Label: JJ

Relation to mother node: Daughter

Figure 2: Searching for a subtree in a monolingual treebank



1690 For its part the Hudson 's Bay Company was troubled *-1 by the approach of American settlement .

1691 As the time *ICH*-6 drew near for the drawing of the British-American frontier by terms of the agreement of 1818 , the comp suspected that the Pembina colony -- its own post and Fort Daer -- was on American territory

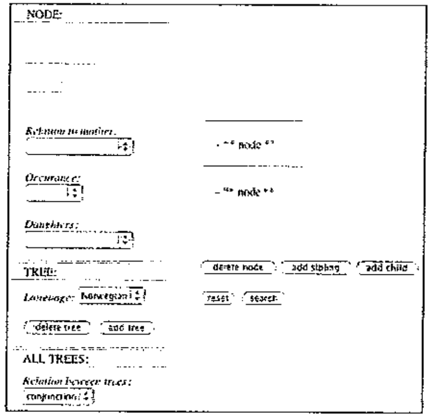
Figure 3: Resulting hits in the Penn Treebank.

The hits are shown in a way that emphasises user-friendliness:

- For each hit, the subtree that matches the searchtree is illustrated with labels and terminals.
- The full sentence is shown below it as a text string, and with the relevant search phrase highlighted.

- The tree for the whole sentence can be viewed by clicking on its left.
- The search expression in TGrep2 query syntax is shown at the results page, providing the user with a way of checking the query, but also to be used as a starting point for a more complex query than the graphical interface itself allows, or even for learning the syntax of TGrep2.

A parallel treebank faces more challenges. First, the same ones apply here as with a monolingual treebank. Second, the treebanks that constitute the parallel treebank may have different tagsets. Third, there may be a series of languages that should be searchable as "source" and "target" corpora at the same time. Fourth, if disjunction is a possible query option, the various combinations multiply and make a user-friendly interface quite hard to maintain. SearchTree has tried to cater for many of the problems. Below is an illustration of the search interface.



Language: Norwegian

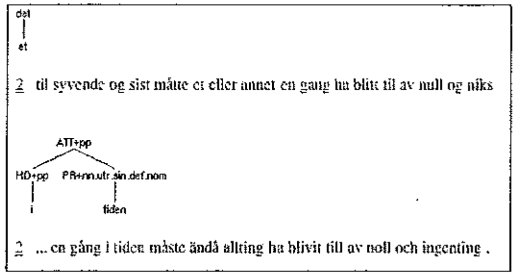
Relation between trees: conjunction

Figure 4: A query interface for parallel treebanks.

As before, the active node is highlighted. In the window above, there are two nodes that can be activated; any search must satisfy both criteria in the languages that are specified for them. For each node, a new node can be added or deleted. The upper left box is for writing the string of a terminal node, and the pop-up menu underneath gives the possibility of specifying parts of the search string, and label. The menus underneath specify the relation the active node has to its mother; optional, obligatory or forbidden; and the relationship it has to its daughters (immediate dominance or other).

The parallel search interface options are seen in the TREE and ALL TREES parts of the window. For each node, a language has to be chosen. The relationship between the trees has to be specified; conjunctive or disjunctive. When a language is chosen, the label menu will reflect its tagset.

Below is part of a results page. The query specified any Norwegian sentence containing a node with the label *det* (determiner) and any sentence-aligned Swedish sentence containing the node PP.



2 til syvende og sist måtte et eller annet en gang ha blitt til av null og niks

2 ... en gång i tiden måste ändå allting ha blivit till av noll och ingenting .

Figure 5: Search results in parallel treebank.

4 Other treebank interfaces

The TIGERin search interface (Voormann and Lezius 2002), based on Tiger Search, is impressive in its expressive power and in its graphics. However, we think that the TIGERin also has some drawbacks that we have solved:

- TIGERin has to be downloaded and installed locally by every user. This can and often does lead to unforeseen, but trivial problems.
- The user must have a local copy of the full treebank.
- TIGERin is highly graphical. However, we think that it is not very intuitive, although this can be a matter of taste. The user is required to click on invisible objects.
- TIGERin presents all tags and categories in ready menus, which is good. However, not all tag names are equally transparent.
- The results from a search with TIGERin are presented as trees of the full sentences in which the search-tree occurs; and with the sentence presented as terminals as a straight line at the bottom of the tree. While this representation has some advantages, such as presenting a full overview of each sentence, the results may be overwhelming. A lot of scrolling is often necessary.

SearchTree avoids some of the problems above for the following reasons: it is web-based; it is totally based on clicking on options that are all visible; the tag names are presented as full names, not abbreviations; the query results are shown in a two step way. First: each hit with the subtree that matches the query, and with the full sentence as a text string with a highlighted search phrase. Second (after optional clicking on the left-hand side): the full sentence tree.

TIGERin makes available the search options of Tiger Search, e.g. for crossing edges, and disjunctive search. SearchTree at the moment is built on top of TGrep2, which makes crossing and secondary edges unavailable for search. This is on the list of future work.

The VIQTORYA query tool (Steiner and Kallmeyer 2002) was developed for the Tübingen German Treebank, and has in common with the current version of SearchTree that it does not cater for crossing and secondary branches. Unlike SearchTree it is based on a tailor-made (first-order logic) query system. VIQTORYA is an abbreviation for "a visual query tool for syntactically annotated corpora", but relies on information that is external to the interface: The annotation scheme must be looked up in specific stylebooks and guidelines that are found elsewhere.

NetGraph (Mirovsky et al, manuscript) and Oraculum (Ljubopytnov et al. 2002) are two systems for searching through the Prague Dependency Treebank. NetGraph, like SearchTree, functions in an Internet environment, and both aim at having graphical interfaces. Oraculum is claimed to be more advanced than NetGraph, but we have not been able to confirm this. Icecup is the search interface for the ICE corpora. It is very advanced w.r.t. queries on trees, but its user interface, although graphical, is not very user-friendly, with a wealth of unintuitive symbols. Unlike SearchTree it must be downloaded on a local computer with specific technical requirements.

5 Conclusion and future work

We have described SearchTree, a user-friendly interface for monolingual and parallel treebank queries. We will continue to increase the flexibility of the interface; increase support for other search engines (at the moment it system works for Microsoft Explorer and Opera); expand the system to support other tree drawing methods; make the system more agnostic to the linguistic approaches to treebank annotation. Especially, we want to support dependency and complex nodes, crossing and secondary branches. In its current form, SearchTree cannot express the full flexibility of the TGrep2 query language. Particularly, it is a hard problem to allow a graphical user interface to express complex bracketing of nodes with disjunction and conjunction conditions in a simple and intuitive way.

References

[1] Icecup: http://www.ucl.ac.uk/english-usage/ice-gb/icecup.htm

[2] SearchTree: http://logos.uio.no/SearchTree

[3]The Sofie Treebank - A Parallel Treebank of North European languages: http://omilia.uio.no/sofie/

[4] The Tiger treebank, search: http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/

[5] TGrep2: http://tedlab.mit.edu/~dr/TGrep2/

[6] Brants, S., S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The Tiger treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories, September 20-21*, Sozopol, Bulgaria.

[7] Ljubopytnov, V., P. Nemec, M. Pilatov, J. Reschke, and J. Stuchl. 2002. Oraculum, a System for Complex Linguistic Queries. In M. Bjelikov (ed.); *SOFSEM 2002 Student Research Forum*, pp. 27-34.

[8] Mirovsky, J., R. Ondruska, and D. Prusa. Searching through the Prague Dependency Treebank – Conception and Architecture. Manuscript. Faculty of Mathematics and Physics. Charles University, Prague.

[9] Rohde, D.L.T. 2004. TGreo2 User Manual version 1.12. http://tedlab.mit.edu/~dr/TGrep2/

[10] Steiner, I., and L. Kallmeyer.2002. VIQTORYA – A Visual Query Tool for Syntactically Annotated Corpora. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas, Gran Canaria, pp. 1704-1711.

[11] Voormann, H., and W. Lezius. 2002. TIGERin – Grafische Eingabe von Benutzeranfragen für ein Baumbank-Anfragewerkzeug. In S. Busemann (ed.); *Proceedings der 6. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2002)*. Saarbrücken.