

Exercise: Searching Treebanks – TIGERSearch and Tregex

1 Task

Get familiar with two tree quering tools, TIGERSearch and Tregex (a java implementation of Tgrep2), by searching for lexical and syntactical information.

2 Getting Started

1. Move in the **directory**: `cd TIGERSEARCH`

2. **Start** the programme: `TIGERSearch &`

(If it doesn't work try: `./TIGERSearch &`. The ampersand '&' allows to run the programme in the background and use the shell for further commands.)

3. Getting **help**:

- (From terminal) `acroread manual.pdf &` (for easy searching)
See especially chapter 12: query language quick reference.
- Built-in help: menu on top of TIGERSearch window (-> query language quick reference)
- Online: http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/manual_html.html

4. **Open** a corpus: menu `top left > Corpus > open`

- open the BROWNSampler click on `DemoCorpora > English > BROWNSampler` (double click)

5. Read the **documentation**: `Documentation > Summary view`

6. Browse **inventory of edge labels**, nonterminal features and terminal features. For an explanation see today's slides on the course webpage.

7. **Explore** the corpus: `top left menu > Corpus > Explore` (Either a TIGERCraphViewer window pops up or it is opened as a icon at the bottom of the screen).

3 Exercise

Graphical Interface

Start with the **graphical interface**: `top of right window > graphical mode`.

(Optional:) See manual for detailed explanations: `top menu > help > Graphical Query editor`

The graphical query editor consists of two regions: the **word level** (at the bottom) and the **non-terminal tree level** (on top)

1. Search for a word

- Click in word region: box pops up (**delete**: right mouse click),

- Double click in inner square: type in 'is' .
- Click on Search (e.g. in right bottom corner).
- Browse the search results in the **TIGERGraphViewer**.
- Try to understand what the difference between matching **graphs** and matching **subgraphs** is.
- **Learn the textual representation** of your query: in Graphical Mode window click on rightmost icon on top ('switch to textual mode')
- **Save textual query as a bookmark**: right mouse click in Textual Mode window > Bookmarks > Add Bookmark to Main Group. Give your bookmark a telling name.

2. Search for word by means of **regular expressions**.

- click on black arrow next to equal sign > 'is a regular expression'
- Type in regular expression: `(isa)s|(a|we)re|be(ing|en)?`
- What does it look for?
- Search.
- Look at the **textual representation of your query**: in Graphical Mode window click on rightmost icon on top ('switch to textual mode').
- Save textual query as a bookmark: right mouse click in Textual Mode window > Bookmarks > Add Bookmark to Main Group. Give your bookmark a telling name.

Textual Mode window

Each node is represented by a pair of square brackets.

Types of search conditions

- Conditions on single node:
 1. Find all words beginning with lower case 'a' that have one of the verbal part-of-speech tags. Save your query as a bookmark.
`[word=/a.* / & pos=/V.* /]`
 2. Find all words NOT beginning with lower case 'a' that have one of the verbal part-of-speech tags. Save your query as a bookmark.
`[word=/[^a.*] & pos=/V.* /]`
- Conditions on two nodes:

basic relations: dominance (>) and precedence (.)

 3. Find all trees that comprise an NP that immediately dominate a proper noun. Save query.
`[cat="NP"] > [pos="NNP"]`

4. Find all trees that comprise an NP that dominate a proper noun.

```
[cat="NP"] >* [pos="NNP"]
```

5. Compare the query results.

6. Find all trees in which 'is' immediately precedes a determiner.

```
[word="is"].[pos="DT"]
```

7. Find all trees in which 'is' immediately precedes a determiner.

```
[word="is"].*[pos="DT"]
```

8. Compare the query results.

- Conditions on more than two nodes: conjunction and variables

- Test for more than one relation: use conjunction '&'

- Refer to the same node more than once: use a variable '#name'

9. Find all trees that comprise an NP that immediately dominates a determiner, an adjective and a normal noun:

```
#1:[cat="NP"] >[pos="DT"] & // "#identifier: []" defines a
                             variable 'identifier'
```

```
#1 > [pos=/JJ.?./] & // "#identifier" refers to content of
                             variable 'identifier'
```

```
#1 > [pos=/NNS?/]
```

Lexical Statistics

All nodes that are referred to by a variable can be analysed quantitatively

10. Define variables for all terminal nodes of the query:

```
#1:[cat="NP"] >#2:[pos="DT"] &
```

```
#1 > #3:[pos=/JJ.?./] &
```

```
#1 > #4:[pos=/NNS?/]
```

- Top left menu > Query > Statistics

- Click in field below Feature 1 > choose #2

- Click in field below #2 > choose word

- Top menu > Add

- Repeat the last steps for feature #3 and #4.

- (Top or bottom menu) > Build

- Inspect the results

- Change the representation: top menu > Frequency and inspect the results.

- Export the statistics: top menu `Export` (choose e.g. text format)

Problem: TIGERSearch does not have an All-quantor.

11. Find verb phrases that contain a verb that is NOT (and not only not immediately) followed by a prepositional phrase (within VP). Look at the query results and try to understand the problem.

Export corpus

Main top left menu > Corpus > Explore

Query > export matches

Exportformat: select: XML piped through XSLT

click on SEARCH

choose BROWNSampler

Export includes: whole corpus

XML piped through XSLT: bracketing format (submit)

In terminal shell: Look at the corpus (`less BROWNSampler`).

Import corpus

In terminal shell: TIGERRegistry &

- Click on TIGERCorpora (such that it is highlighted)
- `Corpus > Insert Corpus`
- Click on `other format`.
- `Corpus ID` (the fill in window might be very small. You can see your typing in the line TIGERXML file below): `Brown-test`
- `Import file > Choose > click on BROWNSampler > Select`
- `Import Filter > Filters available > general Penn Tree Format Filter`
- Keep all the other markings
- Click on `extended indexing`.
- Start
- Window Corpus properties pops up > OK > Close.
- In TIGERSearch > TIGERCorpora > right mouse click > refresh corpus tree

Project Webpage

www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/