

# Doing Surgery on the Penn Treebank with **Tsurgeon**

Corpus-Based Computational Linguistics  
(Chris Brew and Michael White)

2008 Linguistics Summer Mini-Institute

In this activity, we'll see how changes can be made to the trees in the Penn Treebank (PTB) using the Stanford `tsurgeon` tool. Note that since `tsurgeon` doesn't seem to be working through the `tregex` GUI, we'll be using the command line. To actually change files in a corpus, you'd probably want to use the `tsurgeon` API to embed your surgical operations in a simple Java program.

1. Yesterday we saw that relative pronouns are often mistagged as IN instead of WDT. In the first step, we'll find some of the problematic cases and save them in a file. Start `tregex` and load the trees from PTB Section 00, using the `Load trees...` item under the `File` menu. Then, enter a query in the `Pattern:` box that looks for a correctly tagged relative pronoun use of *that*. Note that the WDT should be the child of a WHNP that has a sibling S that dominates a trace. We can find all such nodes using the following query:

```
(__ < that) > (/WHNP/ $ (S << /*T\*/))
```

Now, modify this query to find nodes that are not labeled WDT. Next, save the matching trees to a file using `Save matched trees...` under the `File` menu. The next step will be easier if you save the file in the same directory as the `tregex.jar` file; you might call the file `wrong.WDT.trees`.

2. Next, in the same directory as `wrong.WDT.trees`, make a new file called `relabel.WDT.tsurg`, using your favorite text editor. In the first line of the file, copy your query from step one, and add a label name to the node matching the wrong POS tag (using `=`). Then enter a blank line, then enter a command to relabel the node; supposing you named the mistaken node `whoops`, you'd relabel it as follows:

```
relabel whoops WDT
```

From the command line, you can now run `tsurgeon` as follows (under Windows, use the `.bat` file instead):

```
./run-tsurgeon.command -treeFile wrong_WDT.trees relabel_WDT.tsurg -m
```

The `-m` option shows the before and after results for each match.

3. As another example, let's see how we might simplify expletive-*it* noun phrases. First, search for a NP dominating *it*, then browse for ones that are expletive. You'll see that such NPs have an adjoined S or SBAR that dominates an `*EXP*` trace. Now, write a query that identifies a root NP that has as one child the NP dominating *it*, and as the other child, the S or SBAR dominating the expletive trace. Run the query, and save the matching trees to a file called `expl_it.trees`.
4. To simplify these trees, copy your query as the first line of a file called `simplify_expl_it.tsurg`, then add node names for the root NP, the NP dominating *it*, and the adjoined S or SBAR. Then, after a blank line, add the following operations (using your node names as appropriate):

```
relabel it_np NP-EXPL
delete s_trace
excise root_np root_np
```

From the command line, you can now run `tsurgeon` as follows:

```
./run-tsurgeon.command -treeFile expl_it.trees simplify_expl_it.tsurg -m
```

As an exercise for the ride or flight home, you might think about how you could undo this bit of tree surgery — that is, adjoin an S or SBAR back in (with the appropriate index!).