gDelta: A missing link in the grammar engineering toolchain

Ned Letcher
Tim Baldwin
Rebecca Dridan

THE UNIVERSITY OF MELBOURNE

What is gDelta?

- A tool for use in the grammar engineering process
- Provides feedback on the impact of changes to grammars

Existing tools offer different extremes of feedback:

- very coarse-grained - *ie.* `[incr tsdb()]`
- very fine-grained - *ie.* treebanking

gDelta aims to fill the gap.

What is gDelta?

- Command line Python program
- Outputs static HTML files

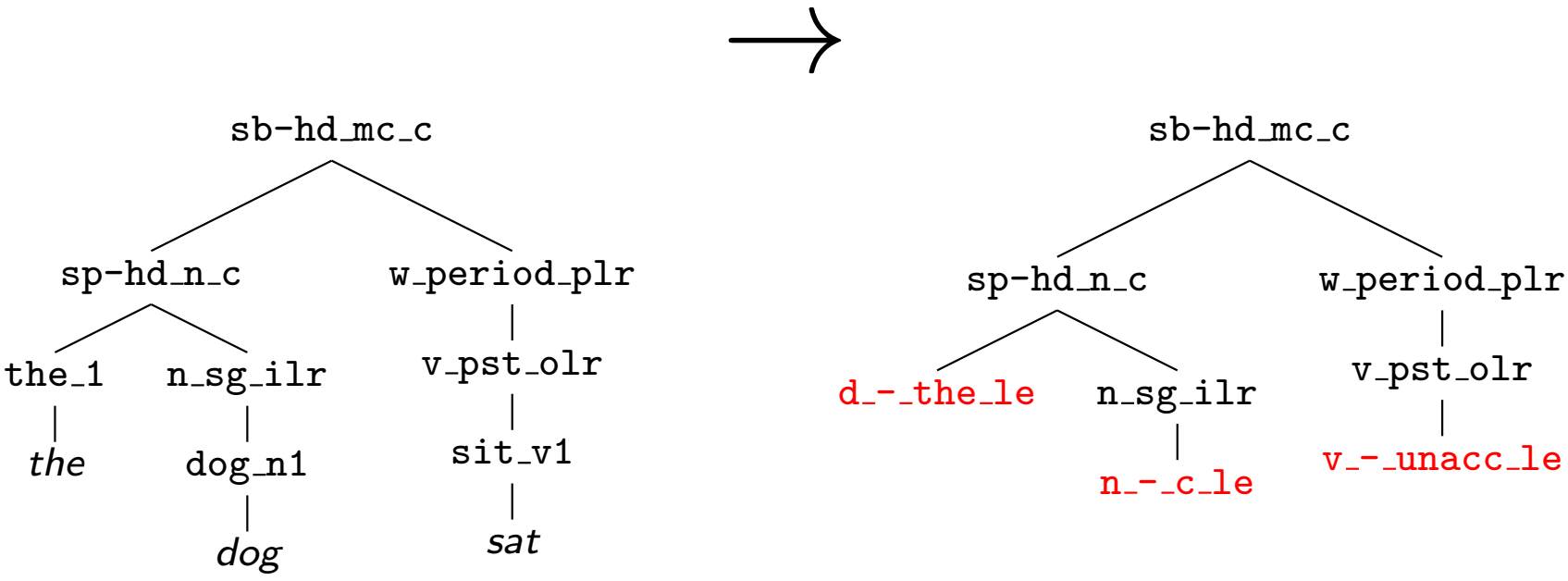
The basic idea:

- two grammars G and G'
- G and G' are both run over the same profile(s)
- gDelta compares parser output from both runs
- reports on changes in features

Like a `diff` tool but for parser output.

Features

- Nodes of the derivation tree
- Lex entries replaced by lex types



gDelta Walkthrough

An attempted change to the ERG from Dan.

- Change to avoid Det-N analysis of *three thirty*.
- Failed to parse good NPs such as in *two days disappeared*.

Running gDelta

Requirements:

- two grammar versions
- grammar entries in `$LOGONROOT/etc/registry`
- tsdb output profiles from the two grammars
- Python 2.6

To run:

```
$ gdelta.py erg ergA ergB ws
```

Give it a try!

SVN: http://svn.nedned.net/gdelta

We are very keen for feedback!

Feature Weighting

- Goal: emphasize changed features
- Problem: change in frequency biases commonly occuring types
- Solution: use change in inverse document frequency (IDF)

$$IDF_{i,G} = log \frac{|P_G| + |P_{G'}|}{1 + |\{p_{i,G} : f_i \in p_{i,G}\}|}$$

$$W_i = |IDF_{i,G} - IDF_{i,G'}|$$

Item Clustering

Idea: use clustering to locate groups of changes

Our approach:

1. convert items into weighted feature vectors
2. cluster using k-means, with $k = \{2,6\}$
3. use Silhouette coefficient to select k
4. select item closest to centroid as exemplar for each cluster