

Woodley Update

- **Mac LUI**
- **ERG Retokenization**
- **Parse Reranking**
- **Unpacking Optimizations**

Mac LUI

- Existing YZLUI for Mac depended on **Carbon** API — deprecated and no longer available as of OS X 10.15
- Interaction paradigm assumed by LUI code is what used to be common for most major OSes, but the replacement API from Apple is a totally different paradigm
- New **MacLUI** app reuses as much YZLUI code as possible (quite large percent)
- Prerelease: <http://sweaglesw.org/linguistics/maclui/>
Caveat: some features still broken — notably Postscript output.

ERG Retokenization

- ERG 2018 and earlier used affixation for punctuation
- Forthcoming release: punctuation as separate tokens, with lowest possible attachment.
- Invalidates 100% of trees in the venerable Redwoods treebank
- Furthermore, the traditional treebank update mechanism is not easily applicable, because chart vertices (used to anchor discriminants) are also different.

ERG Retokenization

- Approach: automatically update the anchoring of the stored decisions, then update as usual.
- Requires computing a mapping from old chart coordinates to new chart coordinates.
- Method: look at the old and new token lattices side by side and align the text
- Categorize symbols in stored decision with respect to punctuation: **above**, **below**, or **at** punctuation level, which then determines which new vertices to use.

Neural Parse Reranking

- Use existing machinery to compute N-best list (10)
- Use neural methods to pick a new winner from those N
- Unsuccessful: tree-shaped network described last year,
 - Including various suggestions, e.g. pretrained embeddings
- Unsuccessful: LSTM over stringified derivations

ACE Unpacking

- Found some low-hanging fruit while unpacking large charts: better data structures in two places where I used to think laziness didn't matter.
- Roughly 30-40% speedup when parsing long sentences (measured on CB + SC01 + SC02 + SC03).
- Regain a small amount of dignity after reranking failures :-)

Time (s) vs. # Words

