Melbourne Site Update

Ned Letcher, Andrew Chester, Tony Wirth and Timothy Baldwin

Talk Outline

1. Supertagging with Hierarchical Tagsets
2. Phenomenal Corpus
3. Dialect Classification
4. Typediff

Supertagging with Hierarchical Tagset: Background

- There has been plenty of work on supertagging for various reasons (robustness, efficiency, ...), but the standard assumption has been that that tagset is "flat"
- With HPSGs, there is, of course, lots of structure to the supertags (= lexical types) that, intuitively, it would appear we should be able to use to good effect
- **Research question:** can we improve the accuracy of supertagging via cleverer user of the type hierarchy?

Supertagging with Hierarchical Tagset: Basic Approach

- Extract type hierarchy from a grammar (focusing exclusively on the ERG for now), and Rebecca-style supertag data
- Also experiment with Penn POS tagging, and shallow-fragmented hierarchy defined by Penn POS tags
- In the first instance focus on supervised learning
- Evaluate in terms of both supertagger accuracy and (ultimately) the impact on parse selection accuracy
- Basic Method: supertag "backoff" using the type hierarchy and trigram HMM (interpolate up the type hierarchy)

Supertagging with Hierarchical Tagset: Details I

- Propagate counts up the lexical type hierarchy to ancestor nodes, and interpolate between child and parent transition probabilities:

$$p_n^{HI} = (1-\alpha) \times p_n^{C} + \alpha \times p_n^{AV}$$

where:

$$p_n^{AV}(t_i|t_{i-n}) = \frac{1}{Z} \sum_{s_{i-n}^i \in \mathcal{P}(t_{i-n}^i)} p_n^{P}(s_{i-n}^i|s_{i-n}^{i-1})$$

where Z is a normalising constant

- Also smoothing of emission probabilities

Supertagging with Hierarchical Tagset: Details II

- And does it work? not terribly well, because:
  - experiments only based on 1-ancestors (which account for only 7% of errors)
  - experiments only based on bigram HMM (expect to get more out of a trigram HMM)

Talk Outline

Basic Idea

- **Phenomenal corpus:** corpus with extent-based markup of a range of phenomena (passive voice, interrogative clauses, imperative clauses, relative clauses and complement clauses)
- **Progress:** a first-version corpus exists, and we are about to start running experiments attempting to automatically identify the extents of different phenomena in text data

Talk Outline

Basic Idea

- **Context:** VarDial 2014 sentence-level classification of dialect/language variant, focusing on AmE vs. BrE
- **Hypothesis:** ERG-based supertagging will help in dialect classification (e.g. *meet Thursday*, *in hospital*)
- **Approach:** use language identification to determine the EN documents, and subclassify using features including supertags

Basic Idea

- **Context:** VarDial 2014 sentence-level classification of dialect/language variant, focusing on AmE vs. BrE
- **Hypothesis:** ERG-based supertagging will help in dialect classification (e.g. *meet Thursday*, *in hospital*)
- **Approach:** use language identification to determine the EN documents, and subclassify using features including supertags
- **Harsh reality:** supertags hinder rather than help, although:
  - unlexicalised supertags
  - some issues with the data

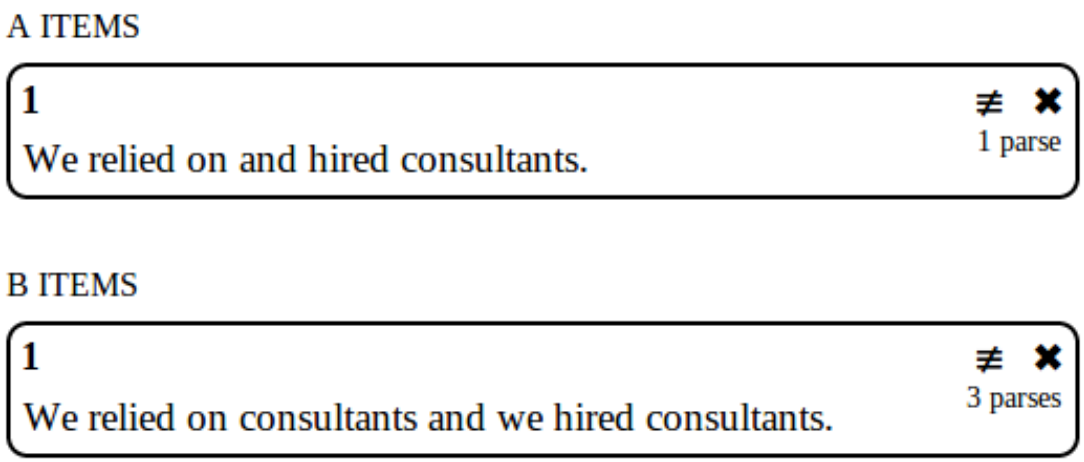
Talk Outline

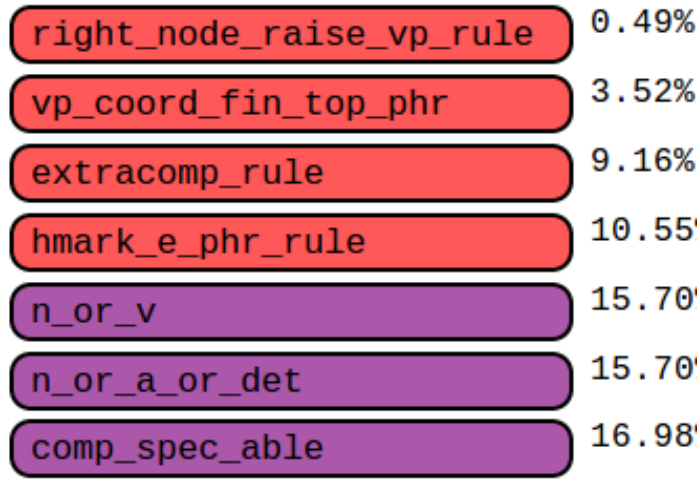
Typediff

- A browser-based tool enabling rapid exploration of all types from the type hierarchy involved in the parsing of input text
- Aim:
  - identify which types from a grammar pertain to the analysis of specific linguistic phenomena
  - contrast phenomenon-positive sentences with similar phenomenon-negative sentences
- Applications:
  - Exploring candidate type signatures for phenomenon detection
  - Phenomenon-based navigation of TDL files
  - Grammar documentation
  - Others?

Methodology

1. Input fragments are parsed with ACE
2. Every type from full AVM of the best parse is extracted
3. Each input fragment is treated as a bag of type names
4. Positive (A) fragments can then be contrasted with negative (B) fragments using set difference
5. Resultant types organized by HPSG theoretic categories (eg `sign`, `synsem`, `head` etc) and sorted by % of gold trees that made use of them





- Other features
  - Fragments can be added if phenomenon can't be captured with minimal pair
  - Relevant tree(s) can be selected from the parse forest
  - Can show differences across inherited supertypes
  - Supports diffing analyses produced by different grammar versions
  - Interfaces with existing DELPH-IN resources
    - Lexical Type Database
    - Fangorn
- Grammars supported
  - ERG, GG, Jacy, HaG, NorSource
  - Requirements: Grammar configured for ACE; any preprocessor runs independently of logon installation
  - Ask Ned if you want your grammar supported
- Homepage: `http://moin.delph-in.net/TypediffTop`
- Live demo