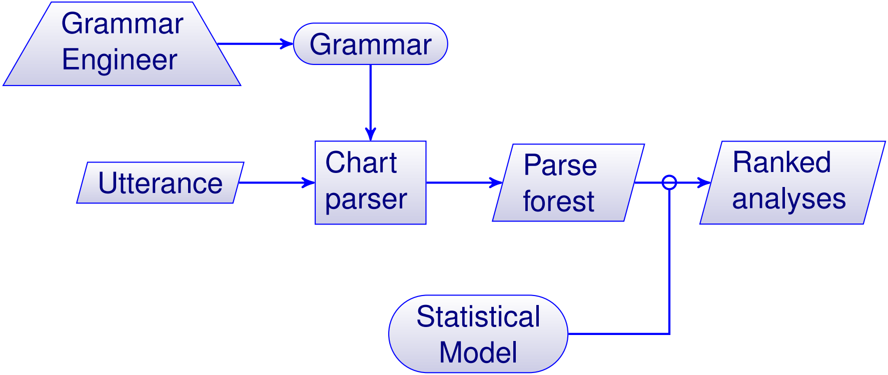
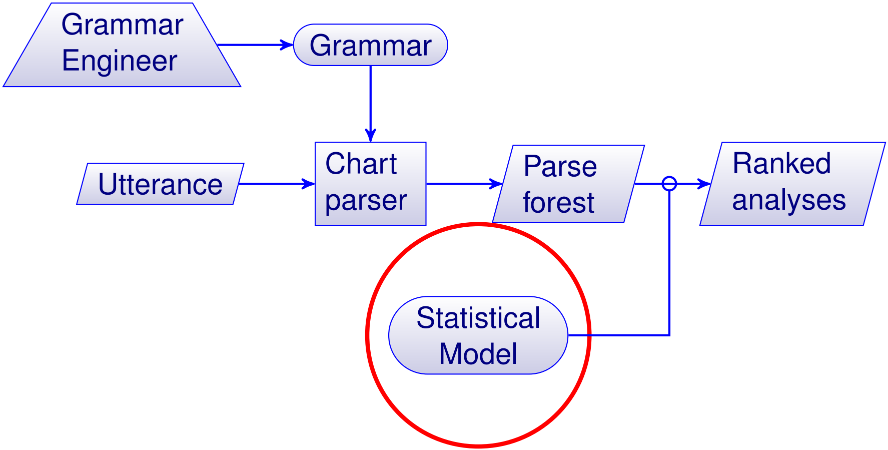
Unsupervised Parse Selection using Supertags

Suquamish, June 2011

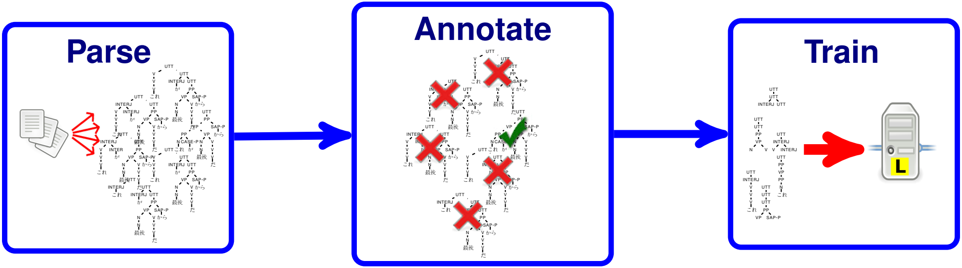
Parsing with DELPH-IN HPSG grammars



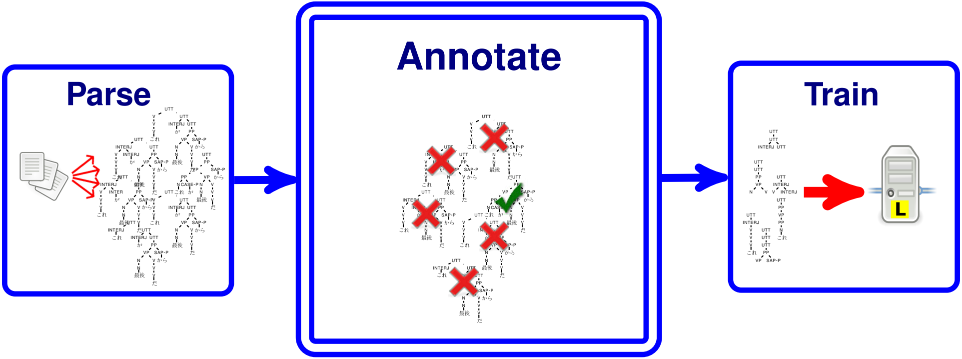
Unsupervised parse selection



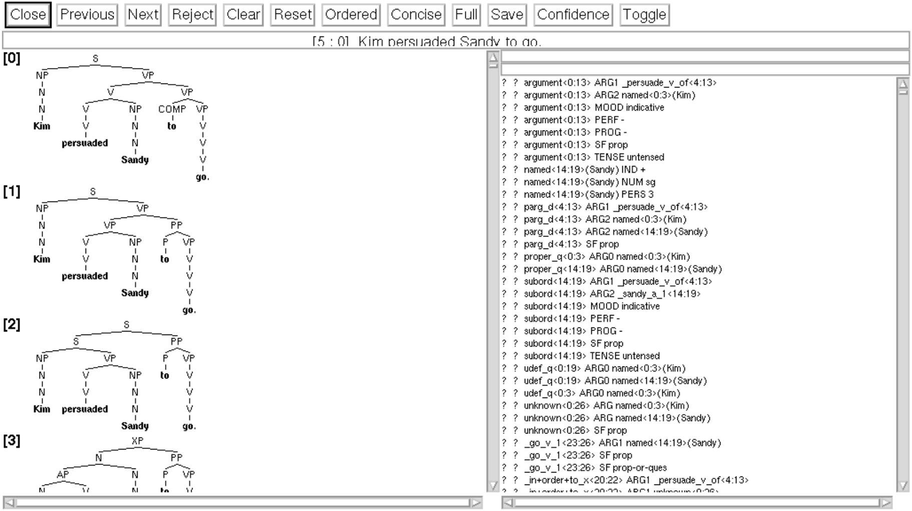
Training a Parse Selection Model



Training a Parse Selection Model



Human Annotation: Treebanking



Human Annotation: Treebanking

- *n* ‘top’ analyses are saved
- treebankers decide between trees by making a series of binary decisions regarding so-called discriminants.
- when the set of trees is reduced to 1, the treebanker can either accept or reject the analysis.
- to be usable for training the parse selection model, an utterance needs:
    - at least one correct analysis
    - at least one *incorrect* analysis

Problems with human annotation

- manual annotation is slow and expensive
- specifically for newer grammars without an existing parse selection model: how do we select the top *n* analyses?
    - set *n* high enough to include all analyses?
      but this makes treebanking very slow
    - randomly select *n* analyses?
      but the correct analysis may not be in the selected set, leading to more rejected utterances and hence less usable training data

Unsupervised parse selection

Can we do better than random selection without requiring human annotations?

Unsupervised parse selection

Can we do better than random selection without requiring human annotations?

We tried three different methods to select pseudo-correct tree (or trees):

- Centroid by edges
- Branching
- Supertags

Heuristic methods

Centroid by edges

For each utterance, chart edges were weighted according to how often they appeared in the parse forest. We marked the tree(s) with the highest average edge weight as ‘correct’.

Heuristic methods

Centroid by edges

For each utterance, chart edges were weighted according to how often they appeared in the parse forest. We marked the tree(s) with the highest average edge weight as ‘correct’.

Branching

For each utterance, the tree(s) with the highest proportion of right- (or left-) branches were marked as ‘correct’.

Supertags

What? Supertags are word categories that carry more information than basic part of speech tags. Specifically, supertags generally encode subcategorisation information.

The tags The supertags we use are lexical types from the grammar:

Examples: v_-_le, n_-_c_le, n_pp_c-of_le, v_np-vp_oeq_le, v_np-np_le, aj_-_i-att_le etc

How? A supertag sequence was predicted for each utterance. We marked the tree(s) that whose leaf types best matched the predicted sequence as ‘correct’.

The tagger

We used a Hidden Markov Model (HMM) tagger, with two different models built using no annotated data:

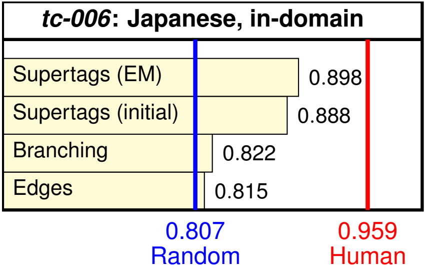
initial We exhaustively parsed sentences under a certain word length, and calculated the emission and transition probabilities from the frequency counts of the lexical types across the full parse forests.

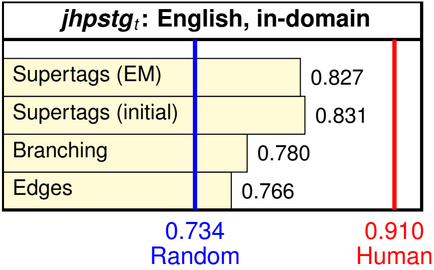
EM Starting from the initial model, we used raw text and the Baum-Welch variant of the EM algorithm to train a new model.

Experiments

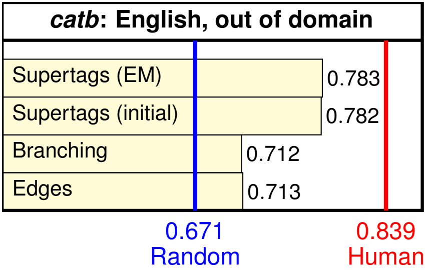
	Language	Sentences	Average words	Average analyses
Training Data	Japanese	6769	10.5	49.6
	English	4855	9.0	59.5
Test Data	Japanese	904	10.7	383.9
	English (in domain)	748	12.8	4115.1
	English (out of domain)	534	17.6	9427.3

Results





Each data set was parsed four times, with the different models trained using the methods described. The graphs show the accuracy of the top-1 parse, measured as the f-score over *Elementary Dependencies*.



Conclusions

- Unsupervised supertagging performs surprisingly well, due to the constrained nature of the HPSG grammars.
- These supertags can then be used to automatically annotate treebank data with sufficient accuracy to produce a parse selection model which is much better than random selection, and about half way towards the accuracy of human annotated data.
- This process is fast, and requires no more than raw text (and a grammar).

Next steps

- Experiment with less mature grammars.
- Measure the impact of using these unsupervised methods to improve the efficiency and accuracy of manual treebanking.
- Look at different unsupervised tagger training methods.
- Explore domain effects arising from the tagger and model training data.

Next steps

- Experiment with less mature grammars.
- Measure the impact of using these unsupervised methods to improve the efficiency and accuracy of manual treebanking.
- Look at different unsupervised tagger training methods.
- Explore domain effects arising from the tagger and model training data.

Experimental Setup with SRG

1. Parse a large amount of data.
2. Train supertagger.
3. Train MaxEnt model using supertag sequences.
4. Train MaxEnt model using right-branching heuristic.
5. Parse items to be tree banked, with each model.
6. Have 2 annotators treebank, shuffling items from each model.
7. Measure inter-annotator agreement, intra-annotator agreement and rejection rates.

Thank You!