PP interactions

Local probabilities only are used by the pre-PRO3Gres parser. Although locality extends further in Dependency Grammar than in constituency grammar (where trees are more nested) and although there are global restrictions in the hand-written grammar, this is a shortcoming. Now: probability that PP2 is a dependent of PP1 (PP1<PP2)

\[
p(\text{verb} < (\text{PP1}, \text{PP2})) = \frac{\#(\text{verb} \in (\text{PP1}, \text{PP2}))}{\#(\text{verb} \in (\text{PP1}, \text{PP2})) + \#(\text{verb} \in (\text{PP2}, \text{PP1}))}
\]

This improves PP-attachment F-score marginally, from 71.9% to 72.4%.

Semantic expectations

The original parser models probabilities using only those syntactic relations that are in competition. For example, in competition with every other relation. A sentence like 

\[
p(R, dist(a, b, c)) = \frac{f(R, dist(a, b, c))}{\sum f(R, dist(a, b, c))}
\]

Sparse data: back-off architecture similar to (Collins and Brooks, 1995), but extending from PP-attachment to most of its dependency relations, and including simple semantic classes from WordNet (Miller et al. 1990). The parser’s label set is close to and can be mapped to GREVAL (Carroll, Minnen, and Briscoe, 2003) and the Stanford scheme (Haverinen et al., 2008). We improve its performance with the following statistical multi-word resources.

Multi-Word Terminology

Our parser Pro3Gres uses chunker pre-processing, it only parses between chunk heads. Multi-word terms (MWT) can be treated like chunks, e.g. by replacing the MWT by its head in a pre-processing step.

- On in-domain text (Penn, Greval):
  - with standard NER (LT-TT2, Grover (2008)): worse to similar, most multi-word terms are shorter than chunks. Re-chunking on term heads leads to similar results.
  - On out-of-domain text (Biomedical):
    - with domain NER: Better than chunker (Weeds et al., 2007), as it corrects many tagging errors, which are frequent (e.g. protein names)
    - with domain-trained tagger: similar to slightly lower performance \(\rightarrow\) statistical > lexical resources

Self-Training

Self-training can improve results where sparseness is worse than error rate. Pro3Gres has a strong correlation between backoff level and parser accuracy. Fully lexicalized decisions have much higher performance than those further down the back-off chain.

\[
\begin{align*}
\text{Level 0: head + preposition + description} \\
\text{Level 1: verb + preposition} \\
\text{Level 2: class + preposition + noun} \\
\text{Level 4: head class + preposition + description-noun class} \\
\text{Level 5: preposition + description-noun class} \\
\text{Level 6: preposition only}
\end{align*}
\]

We need methods to reduce sparseness \(\rightarrow\) more decisions can be taken at early backoff levels.

Distributional Semantics

Non-negative matrix factorization (Lee and Seung, 2001) is a vector modeling technique, but unseem combinations. It never uses negative weights \(\rightarrow\) suitable for treating probabilities

For our verb/noun-PP attachment probability matrix

- Initial verb-prep and noun-prep matrices are filled with attachment probabilities. Null counts are given \(p=0.2\)
- We use a version for multiple PP's (verb-prep1-prep2 : noun-prep1-prep2)
- The approximated matrix contains non-sparse (non-zero) probabilities for every verb/noun-prep1-prep2 combination.

This improves F-score from 71.9% to 72.4%.

Combined Model

We combine the described improvements. F-Score increases from 71.9% to 72.9%. This is modest, but the upper bound is low due to

- lemmatization, tagging and chunking errors
- mapping to gold standard representation, e.g. grammar assumptions

We also tried many lexical resources, but got no improvement \(\rightarrow\) implicit in stats

References


