Parser lexicalisation through self-learning

Marek Rei & Ted Briscoe
Computer Laboratory, University of Cambridge, UK

BACKGROUND

- The use of lexically-conditioned features, such as relations between lemmas or word forms, is important for creating accurate parsers.
- However, utilising such features leads the parser to learn information that is often specific to the domain and/or genre of the training data. Furthermore, manual creation of in-domain treebanks is expensive and time-consuming.
- Unlexicalised parsers avoid using lexical information and select a syntactic analysis using only more general features, such as POS tags. While they cannot be expected to achieve optimal performance when trained and tested in a single domain, unlexicalised parsers can be surprisingly competitive with their lexicalised counterparts.
- Instead of trying to adapt a lexicalised parser to new domains, perhaps we can directly integrate lexical features with any unlexicalised parser.

HYPOTHESIS

- We hypothesise that a large corpus will often contain examples of dependency relations in non-ambiguous contexts, and these will mostly be correctly parsed by an unlexicalised parser.
- Lexical statistics derived from the corpus can then be used to select the correct parse in a more difficult context.
- This would allow the unlexicalised parser to learn lexical features directly from its own output, without any manual annotation.

EXAMPLES

Interest can be both noun or a verb, which can lead to ambiguous sentences:
- Government raises interest rates
- Government projects interest researchers

After learning from non-ambiguous cases, we could infer that interest is likely to modify rates:
- Interest rates are increasing
- Government projects receive funding

SYSTEM OVERVIEW

1. A large corpus of in-domain text is parsed with the unlexicalised parser.
2. New edges are added to dependency graphs, to model selected higher-order dependencies.
3. Maximum-likelihood probabilities are found for all lexical relations.
4. New confidence scores are calculated for alternative parses of each sentence.
5. Parses are re-ranked, improving the accuracy of the top parse.

GRAPH SCORING

Edge scores are combined into graph scores by first averaging over all edges for each node, and then over all nodes.

\[ \text{NodeScore}(n) = \frac{\sum_{e \in E(n)} \text{EdgeScore}(e) \times \text{isDep}(e, n)}{\sum_{e \in E(n)} \text{isDep}(e, n)} \]

\[ \text{GraphScore}(g) = \frac{\sum_{n \in N(g)} \text{NodeScore}(n)}{|N(g)|} \]

EXPERIMENTS

We make use of the unlexicalised RASP parser (Briscoe, 2006) as the baseline system. Experiments were performed on Wall Street Journal (DepBank/GR) and biomedical (Genia-GR) datasets.

RELATION SCORING

Every edge in the modified graph is assigned a confidence score, using various methods:
- The probability of edge \( e \) belonging to the best possible parse, based on ranking from the unlexicalised parser:
  \[ \text{RES}(e) = \frac{1}{\sum_{e \in E} \text{contains}(g', e)} \]
- The probability of a specific relation type occurring between two words, given that the words are seen in a sentence together, calculated from the background corpus:
  \[ \text{CES}_2(w_2) = \frac{P(\text{rel}, w_1, w_2)}{P(w_1, w_2)} \]
- Smoothing these probabilities with distributional similarity:
  \[ \text{CES}^\ast_2(w_2) = \frac{\sum_{c \in C} \text{sim}(c, w_1) \times P(\text{rel}(w_1, w_2))}{\sum_{c \in C} \text{sim}(c, w_1)} \]
- Combining together all alternative scoring methods.

SUMMARY

- The unlexicalised parser is able to learn lexical features from its own output.
- Significant improvement in F-score achieved on both WSJ and Genia data.
- The method managed to close more than half of the gap between the performance of a fully-supervised in-domain lexicalised parser and a weakly-supervised unlexicalised one.
- The framework requires only a large corpus of in-domain text. No manual annotation or supervised training is needed.