Characterizing the Errors of Data-Driven Dependency Parsing Models

McDonald & Nivre 2007

Background

Two basic approaches to dependency parsing are all pairs and stepwise.

- All pairs = graph based
- stepwise = transition based

"All pairs" approaches make decisions globally, use exact inference but have relatively impoverished features

"stepwise" approaches make greedy decisions, but have a rich feature representation including past decisions.

Both achieve similar performance but the kinds of errors they make are different. Segue and Lavie (2006) shwo that combining the predictions of both types of models yields "significantly improved accuracy" This paper is going to talk about the strengths and weaknesses of the approaches.

Two models for dependency parsing

Preliminaries

Describes what a dependency tree is, & graph based and transition based dependency parsing.

Overall, Kuebler et al (2009) has a more thorough discussion of the different approaches.

Global Graph based parsing

Labels are portrayed as part of the scoring function in this work. *I believe how scoring labels works varies between different approaches but I have to look further into this*

> The primary disadvantage of these models is that the feature representation is restricted to a limited number of graphs arcs. This restriction is required so that both inference and learning are tractable

MSTparser is the implementation used.

Local, Greedy, Transition-Based Parsing

> The primary advantage of these models is that afeatures are not restricted to a limited number of graph arcs but can take into account the entire dependency graph built so far. The main disadvantage is that the greedy parsing strategy may lead to error propogation.

CONLL-X shared task

13 languages 19 systems labeled attachment score was official metric (percentage of tokens, excluding punctuation, that are assigned botht he correct head and the correct dependency label).

Error analysis

Graph factors

All current parsers have more trouble on longer sentences. MaltParser performs better in shorter sentences, worse as sentences get longer. Attributed to likelihood of error propogation being higher for longer sentences and richer feature representation as beneficial for short sentences.

MSTParser far more precise than MaltParser for longer dependency arcs (where the length is the length of the predicted arc). MaltParser does better for shorter dependency arcs. Overall MSTParser is not affected by dependency length.

MSTParser is far more precise close to the root and is less precise then Malt further from the root.

Dependency arcs further from the root are (usually) created first in transition based systems. Thus this is further evidence that error propogation is partly to blame for the difference between the two approaches.

> MSTParser over predicts arcs near the bottom of the graph. Whereas MaltParser pushes difficult parsing decisions higher in the graph, MST Parser appears to push these decisions lower

Linguistic Factors

Findings with regard to part of speech associations are tied to previous findings of position in graph.

Adpositions are a bit strange because they have high average root distance and low average dependency length but MSTParser does okay on them.

Revision #5

Created Thu, May 7, 2020 4:33 PM by kenneth

Updated Thu, May 7, 2020 5:17 PM by kenneth