Dependency parsing

TA 7

Progress in Dependency parsing

http://nlpprogress.com/english/dependency_parsing.html

Methods in Dependency parsing

1. **Dynamic programming**

   Eisner (1996) gives a clever algorithm with complexity O(n3), by producing parse items with heads at the ends rather than in the middle

2. **Graph algorithms**

   You create a Minimum Spanning Tree for a sentence McDonald et al.’s (2005) O(n2) MSTParserscores dependencies independently using an ML classifier (he uses MIRA, for online learning, but it can be something else) Neural graph-based parser: Dozat and Manning (2017) et seq. –very successful!

3. **Constraint Satisfaction** Edges are eliminated that don’t satisfy hard constraints. Karlsson(1990), etc.
4. **“Transition-based parsing” or “deterministic dependency parsing”** Greedy choice of attachments guided by good machine learning classifiers E.g., MaltParser(Nivreet al. 2008). Has proven highly effective. And fast.

Biaffine parser

A biaffine parser uses transformers for representing input sentences, with no other feature. The graph parser is a semantic parser that exploits a similar architecture except for using a sigmoid crossentropy loss function to return multiple values for the predicted arcs.

Deep Biaffine Attention for Neural Dependency Parsing

https://arxiv.org/abs/1611.01734

Preprocessing and Format conversions in Treebanks

https://github.com/hankcs/TreebankPreprocessing

Some more tools

https://github.com/UniversalDependencies/tools

Universal Dependencies

Manning's Law:

- UD needs to be satisfactory for analysis of individual languages.
- UD needs to be good for linguistic typology.
- UD must be suitable for rapid, consistent annotation.
- UD must be suitable for computer parsing with high accuracy.
- UD must be easily comprehended and used by a non-linguist.
- UD must provide good support for downstream NLP tasks.

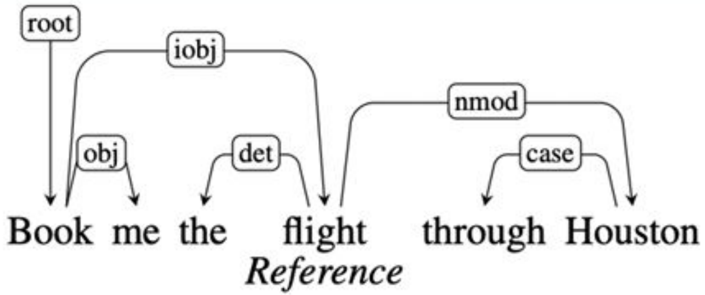
Evaluation

- Unlabeled attachment score (UAS)

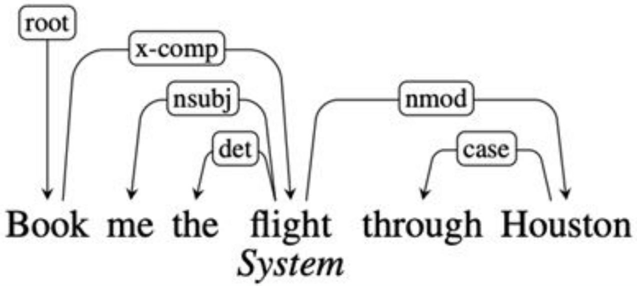
  = percentage of words that have been assigned the correct head

- Labeled attachment score (LAS)

  = percentage of words that have been assigned the correct head & label



Reference



System

UAS = ? LAS = ?

Refs

https://www.cs.princeton.edu/courses/archive/fall19/cos484/lectures/lec11.pdf

Implementation example 1

https://www.cse.chalmers.se/~richajo/nlp2019/l7/Biaffine%20dependency%20parsing.html

Pretrained example (udify)

https://github.com/hyperparticle/udify

Implementation example 2

https://github.com/Unipisa/biaffine-parser

Example 3 - Allennlp

https://paperswithcode.com/model/deep-biaffine-attention-for-neural-dependency

https://storage.googleapis.com/allennlp-public-models/biaffine-dependency-parser-ptb-2020.04.06.tar.gz

Easier implementation example

https://nlp.gluon.ai/model_zoo/parsing/index.html