

Dependency Parsing - Hindi

Ayan Biswas - 2019121009

Presentation video [here](#).

Veeral Agarwal - 2019114009

Mentor - Saujas Vaduguru

Course instructor - Prof. Manish Shrivastava

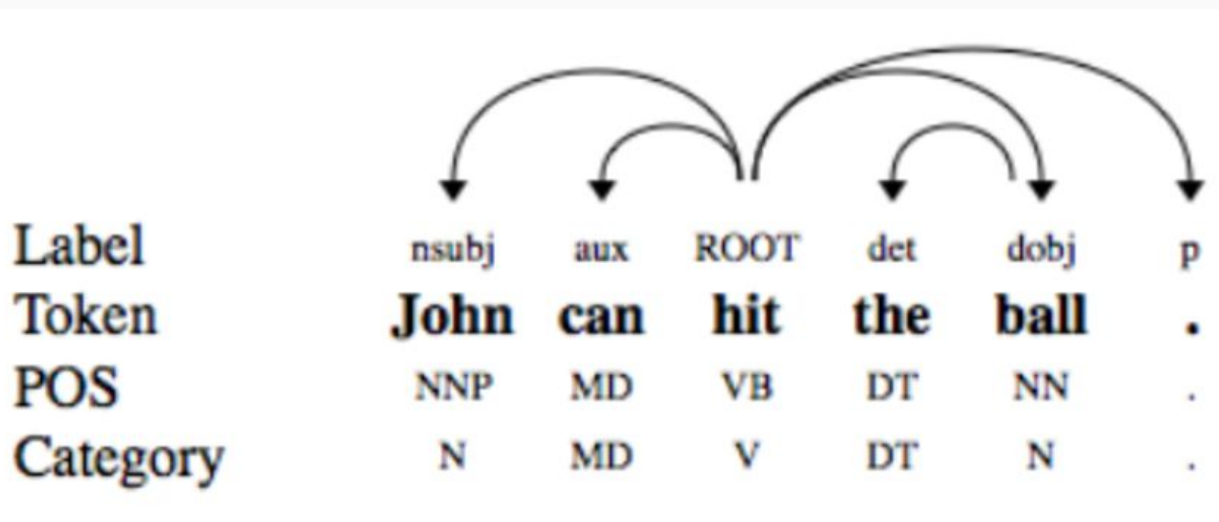


Problem Statement

Dependency parsing - Hindi

In this project, we are creating a dependency parser (hindi) for which the training dataset will be provided. A dependency parser identifies the syntactic dependency between words in a sentence. While a lot of literature exists in the dependency parser community in NLP, there are few well developed tools for indian languages.

Introduction



Algorithm

The initial state is to have all of the words in order on the buffer, with a single dummy ROOT node on the stack. The following transitions can be applied:

LEFT-ARC: marks the second item on the stack as a dependent of the first item, and removes the second item from the stack (if the stack contains at least two items).

Algorithm (contd)

RIGHT-ARC: marks the first item on the stack as a dependent of the second item, and removes the first item from the stack (if the stack contains at least two items).

SHIFT: removes a word from the buffer and pushes it onto the stack (if the buffer is not empty).

Method

Greedy Transition/Shift Parsing - Greedy choice to be done using a Learning Algorithm

- SVMs
- Neural networks

Features?

- Template based - Transition-based Dependency Parsing with Rich Non-local Features
- Concept Features: word i + pos i + psp i + word j + pos j + psp j
- Concept Labels: L - Left Arc / R - Right Arc

Some References:

- <https://www.aclweb.org/anthology/D14-1082.pdf> - Dependency parsing using neural networks for the greedy choice using feature templates.
- <https://www.aclweb.org/anthology/W12-3410.pdf> use of svms for dependency parsing in Tamil using model agreement features.
- <https://web.stanford.edu/~jurafsky/slp3/14.pdf> . Speech and Language Processing. Daniel Jurafsky & James H. Martin.

Copyright © 2020. All rights reserved. Draft of December 30, 2020. Chapter - 14

Method (cont)

- SVMs are comparatively simple, we can take a feature vectors and use kernels.
- Neural Nets would require us to take a look at behaviour of a few different architectures.
- Yet to be finalized.

Dataset

[Universal Dependencies Hindi Tree Bank](#)

Interim Timeline

- Interim: Setup data preprocessing and Feature-Extraction pipeline. Implement a parser with greedy choice as a swappable module.
- Final: After testing various approaches, arrive at a robust model for Dependency Parsing in Hindi.

Thanks!