

# Homework 03: Transition-based Parsing

CS 6501-005 Natural Language Processing

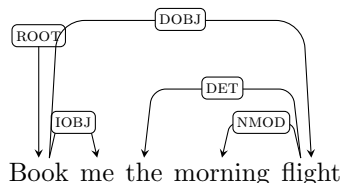
Updated on: October 19, 2019

1. **Extracting Parsing Actions.** One way to obtain a transition-based parser is to treat parsing as a classification problem. At time step  $t$ , the parsing action is predicted by considering the status of the input buffer and the stack. Specifically, the decision function used by a parser can be defined as

$$\hat{y}_t = \operatorname{argmax}_{y \in \mathcal{Y}} \boldsymbol{\theta}^\top \mathbf{f}(\mathbf{x}_t, y) \quad (1)$$

where  $\boldsymbol{\theta}$  is the classification parameter,  $\mathcal{Y}$  is the set of candidate parsing actions, and  $\hat{y}_t$  is the predicted parsing action.  $\mathbf{x}_t$  is the numeric representation of the input buffer/stack status, usually extracted from the top two words on the stack and the head word in the input buffer.

To train a classifier as in Equation 1, we need the training example as a collection of pairs  $\{(\mathbf{x}_t, y_t)\}$ . It means we have convert a parsing tree as following to a sequence of parsing actions. For simplicity, in this problem, we can ignore the dependency relations and only focus on three basic parsing actions. Please refer to lecture 11 slides 19 for the list of parsing actions related to this example.



- (a) (3 points) Please design an algorithm that can simulate the parsing procedure based on a given dependency tree and implement it. Your answer should include (i) a brief description of the algorithm and (ii) an implementation with the file name `[ComputingID]-alg.py`.

Note that, there are some special cases during parsing, for example, the first two actions must be SHIFT; when the input buffer is empty, SHIFT action is illegal, etc.

- (b) (3 points) In the attached `data.conll` file, you will find 505 sentences with annotated dependency tree in the CoNLL format. In each section of the data, please ignore the lines starting with `#`. To get the dependency tree, you need following three columns
- Column 1: word index
  - Column 2: word
  - Column 7: head word index

Your answer should include a file of parsing actions extracted from the 505 example sentences, with file name `[ComputingID]-parsing-actions.txt`.

2. **Constituent Parsing.** Transition-based parsing can also be used to build a parse tree based on the CFG grammar, as discussed in lecture 9 and lecture 10.

- (a) (2 points) Design a set of parsing actions that can build any parse tree from the following grammar.

S	⇒	NP	VP	1.0
VP	⇒	Vi		0.4
VP	⇒	Vt	NP	0.4
VP	⇒	VP	PP	0.2
NP	⇒	DT	NN	0.3
NP	⇒	NP	PP	0.7
PP	⇒	P	NP	1.0

Vi	⇒	sleeps	1.0
Vt	⇒	saw	1.0
NN	⇒	man	0.7
NN	⇒	woman	0.2
NN	⇒	telescope	0.1
DT	⇒	the	1.0
IN	⇒	with	0.5
IN	⇒	in	0.5

In your answer, explain carefully the meaning of each action and how it can be done with basic operations associated with input buffer and stack. As an example of defining parsing actions, please refer to the slides of pages 13 - 15 in lecture 11.

- (b) (2 points) Use the parsing actions defined in (2) to reconstruct the parsing action sequence of the following tree

