Syntax Parser Unit Testing Evidences

# Penn Treebank English Dataset

The input dataset is of the form syntax trees. These trees need to be transformed to an interpretable form such that we know in which level(iteration), which set of words can be combined into a single composition node.

## No sample is missing after dataset processing:

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **No. of Records in raw dataset** | **No. of records after processing** | **Status** |
| 02-21.10way.clean.txt- (train) | 39832 | 39832 | Badge Tick1 with solid fill |
| 22.auto.clean.txt (validation) | 1700 | 1700 | Badge Tick1 with solid fill |
| 23.auto.clean.txt (test) | 2416 | 2416 | Badge Tick1 with solid fill |

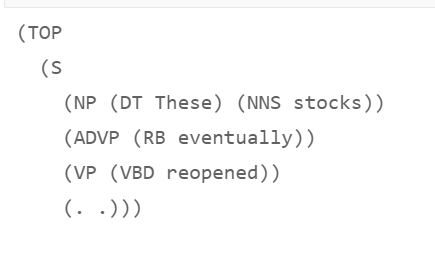


## The transformed syntax tree should not miss any words/token from the raw syntax tree:

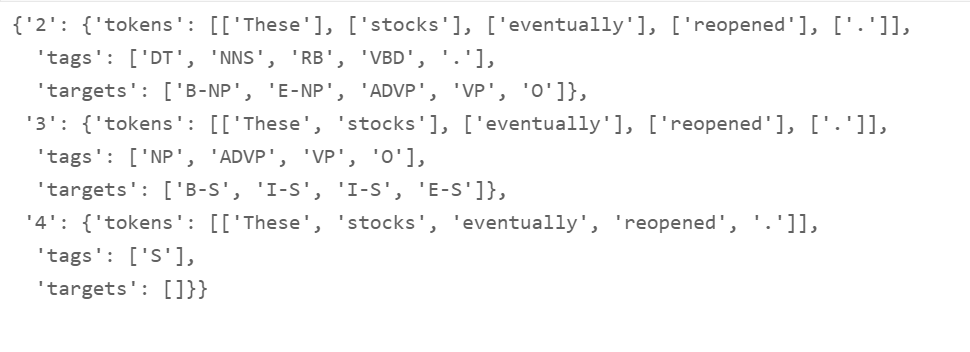
Let us examine the following examples:

**Example 1:** “(TOP (S (NP (DT These) (NNS stocks)) (ADVP (RB eventually)) (VP (VBD reopened)) (. .)))”

**Tree Parsing output:**

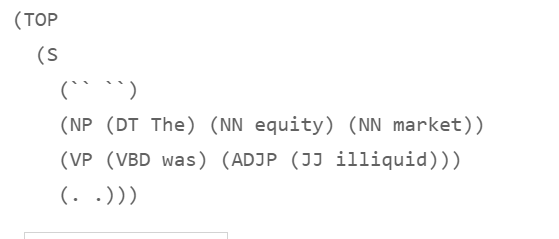


**Transformed json output:**

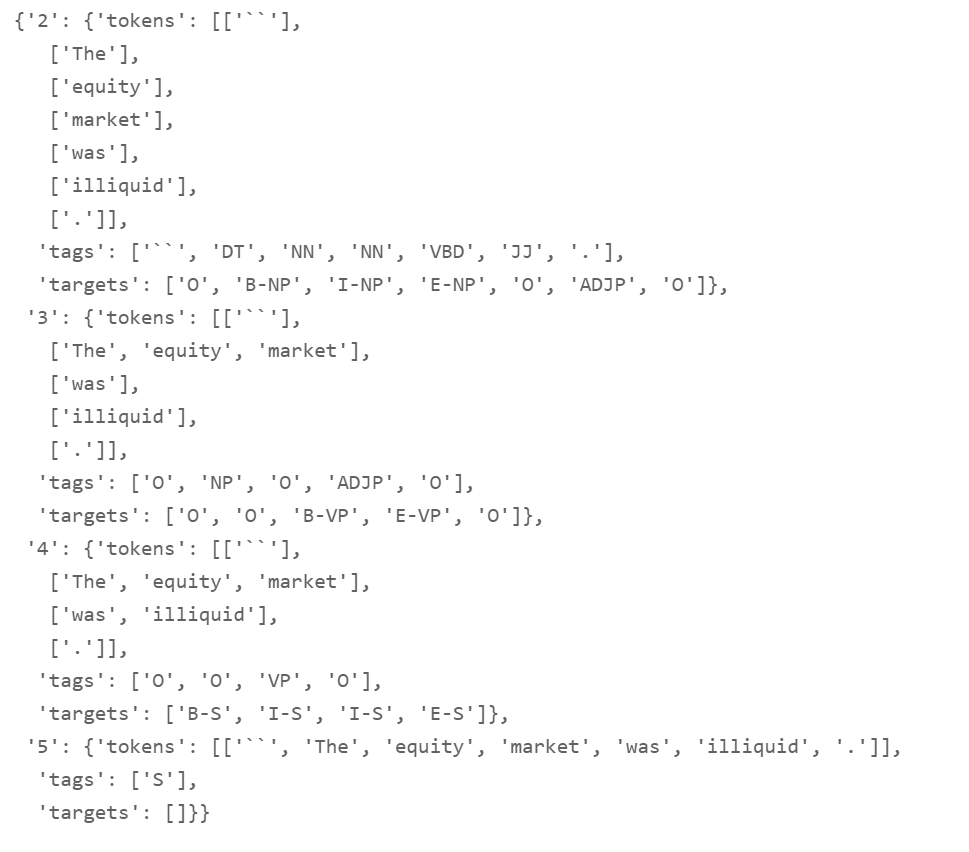


**Example 2:** '(TOP (S (`` ``) (NP (DT The) (NN equity) (NN market)) (VP (VBD was) (ADJP (JJ illiquid))) (. .)))'

**Tree Parsing output:**



**Transformed json output:**



# Dictionary (Word W, POS Tags P and Targets T)

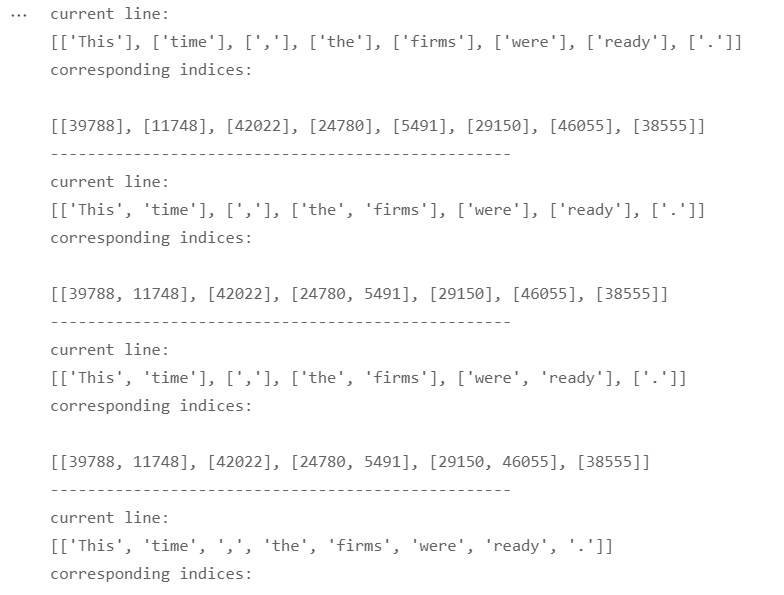
**Details:**

* Word lookup tables: tokens\_lkp.pkl and tokens\_lkp\_rev.pkl
* POS Tags lookup tables: tags\_lkp.pkl and tags\_lkp\_rev.pkl
* Targets(BIOES) lookup tables: targets\_lkp.pkl and targets\_lkp\_rev.pkl

# PyTorch Dataset:

## Tokens -> Index -> Tensors

* Tokens to idx pipeline: pass



* Tags and Targets to idx pipeline:

