Rule Based Constituency Parse to Dependency Parse and Dependency Parse to Constituency Parse

Karan Agarwalla 180050045

Rishi Agarwal 180050086

Sumanyu Ghoshal 180070060

Abstract

The task at hand is to use rule-based models to convert one kind of parsing to another. In this assignment, we show our method of creating the model, and compare our model with the Stanford Models which is also available with our code, and AllenNLP models available online.

1. Constituency Parse To Dependency Parse

For converting the Constituency Parse to Dependency Parse, we divide the task into 2 parts:

A. Generating the Dependency Parse Tree
   The Dependency Parse Tree is generated via the following steps:
   - Use recursion from smaller to larger phrases.
   - Find the head for the phrase in question.
   - Use the heads of the children to find the head of larger phrases.
   - Head Rules:
     - For Noun Phrase:
       - Pick the last noun from the top level nouns
       - If no nouns in top level, pick the last noun phrase from the top level noun phrases and find the head of this noun phrase
       - If no noun phrases, pick the last noun in the entire phrase
       - If no nouns in the phrase, pick the last word of the phrase
     - For Verb Phrase:
       - Pick the first verb from the top level verbs
       - If no verbs in top level, pick the first verb phrase from the top level verb phrases and find the head of this verb phrase
       - If no verb phrases, pick the first verb in the entire phrase
       - If no verbs in the phrase, pick the first word of the phrase
     - For any other Phrase:
       Note: Stanford dependency parsing generates trees which carry semantic meaning. So, we also tried to create rules which would give similar output as the Stanford dependency parser.
       - Pick the head of the top level verb phrases, if any
       - If no verb phrases in top level, pick the head of the top level noun phrases, if any
       - Else pick the head of top level S-phrase

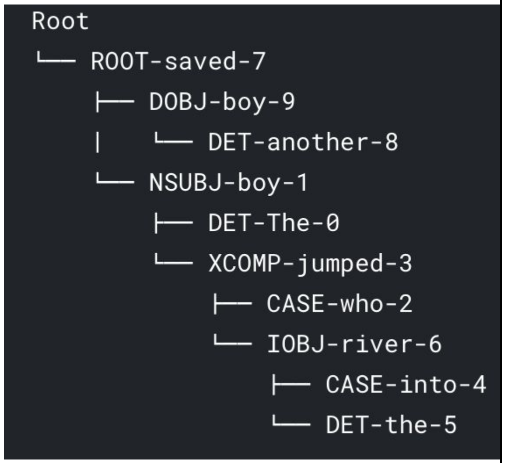
B. Labelling the Dependency Parse
   We have used the following rules to find the labels:
   - **Determiner**: DT tag in Constituency parse
   - **CC**: Conjunction tag from Constituency parse
   - **Conj**: Use CC as child to tag conj
   - **CASE**: any not verb/noun/adjective/det/
   - **NMOD**: (nn, nn) for parent-child with Preposition in between
   - **AMOD**: (nn, adj) for parent-child
   - **NUMMOD**:(nn, num) for parent-child
   - **APPOS**: (nn, nn) for parent-child with Preposition in between
   - **NSUBJ**: Noun to the left of main verb

- **DOBJ**: Latter noun after the verb
- **IOBJ**: First noun after verb
- **CCOMP**: (verb, verb) or (verb, adjective) for parent-child
- **XCOMP**: (verb, verb) or (verb, adjective) for parent-child, when sentences without both object and subject & is not linked to the main verb. Also assigned when no tag can be assigned to a word

Error Analysis

- The child to the root is assumed to be the main verb. This might not be the case in a few examples. Like, in the case of the sentence: ***The children played basketball.***, the word ‘basketball’ is taken as the child to the root,whereas as per our model ‘played’ would be the child to the root.
- In the case of Tagging, we face issues when handling the CCOMP and XCOMP.
- We have not been able to support the *acl* tag as in the paradigms for the rules that we have made the rules would turn ambiguous.

Example (Incorrect):
Sentence: The boy who jumped into the river saved another boy.

Dependency Parse Using our Model	Dependency Parse (Using Stanford Model)
	[(('saved', 'VBD'), 'nsubj', ('boy', 'NN')), (('boy', 'NN'), 'det', ('The', 'DT')), (('boy', 'NN'), 'acl:relcl', ('jumped', 'VBD')), (('jumped', 'VBD'), 'nsubj', ('who', 'WP')), (('jumped', 'VBD'), 'obl', ('river', 'NN')), (('river', 'NN'), 'case', ('into', 'IN')), (('river', 'NN'), 'det', ('the', 'DT')), (('saved', 'VBD'), 'obj', ('boy', 'NN')), (('boy', 'NN'), 'det', ('another', 'DT'))]

```
Root
└── ROOT-saved-7
    ├── DOBJ-boy-9
    |   └── DET-another-8
    └── NSUBJ-boy-1
        ├── DET-The-0
        └── XCOMP-jumped-3
            ├── CASE-who-2
            └── IOBJ-river-6
                ├── CASE-into-4
                └── DET-the-5
```

2. Dependency Parse to Constituency Parse

For converting the Dependency Parse to Constituency Parse, we have referred to the paper:http://cdn.iiit.ac.in/cdn/ltrc.iiit.ac.in/hutb_release/related_publications/Towards_a_Multi-Representational_Treebank.pdf.
We further divided the problem into 2 steps:

A. Generating the Tree:
    a. Generate the phrase (shuffled words though based on the inorder traversal of the DP) to create left and right children based on the index of words.
    b. Generate phrase trees for children.
B. Labelling:
    a. The PoS tags for each word are already available in the dependency parse.
    b. The phrase tag has been obtained based on the PoS tag of the head of the dependency tree.

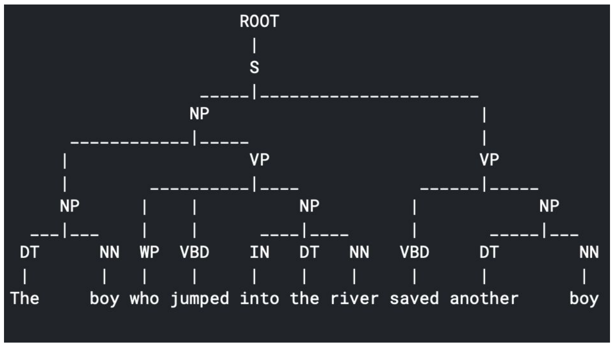
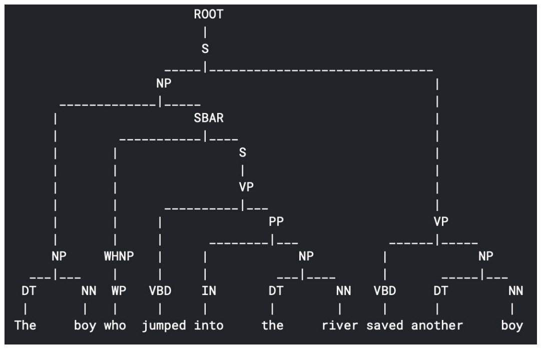
Error Analysis

We are unable to support the **tagging of** (while generating the correct tree structure):

- SBAR
- PP

Example:

Sentence: The boy who jumped into the river saved another boy.

Constituency Parse Using our Model	Constituency Parse Using Stanford Parser
	

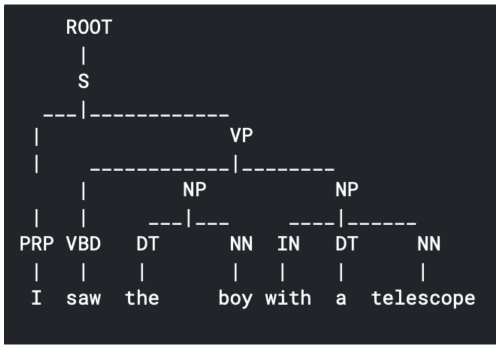
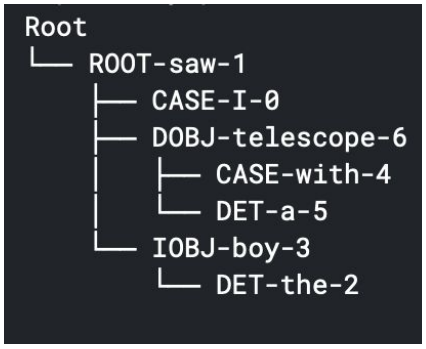
```
                       ROOT
                        |
                        S
                   -----|-----------------------
                  NP                           |
      ------------|------                      |
     |                  VP                     VP
     |        ----------|----           ------|-----
     NP      |    |          NP          |            NP
  ---|---    |    |      ----|----      |       -----|---
 DT      NN  WP  VBD    IN   DT   NN   VBD     DT        NN
 |       |   |    |     |    |    |     |      |         |
The     boy who jumped into the river saved another     boy
```

```
                              ROOT
                               |
                               S
                          -----|------------------------------
                         NP                                  |
           --------------|-----                              |
          |                  SBAR                            |
          |        ------------|----                         |
          |       |                 S                        |
          |       |                 |                        |
          |       |                 VP                       |
          |       |        ----------|---                    |
          |       |       |             PP                   VP
          |       |       |       -------|---          ------|-----
          NP     WHNP     |      |          NP         |           NP
       ---|---    |       |      |       ---|----      |      -----|---
      DT     NN   WP     VBD     IN      DT      NN   VBD     DT       NN
      |      |    |       |      |       |       |     |      |        |
     The    boy  who   jumped  into     the    river saved another   boy
```

As we see in the trees generated, while the trees itself are correct, the tagging is incorrect as rules for SBAR and PP while tagging can make the rules for NP to be ambiguous.

3. Some Additional Examples

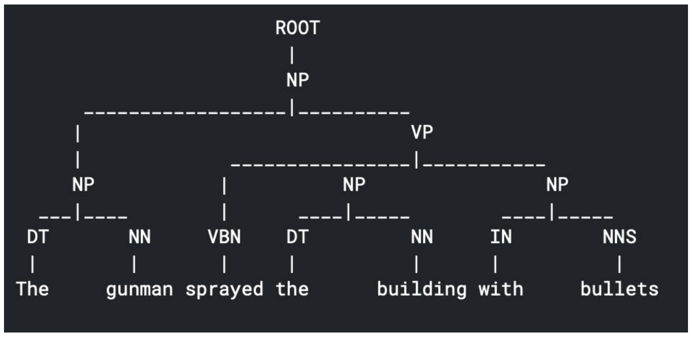
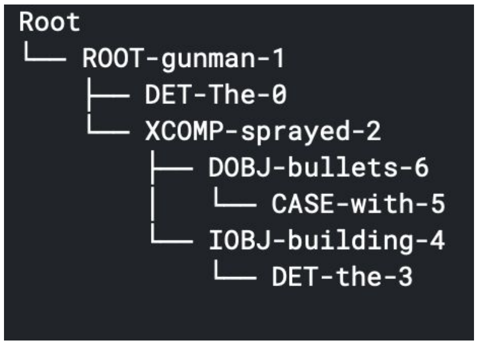
Example 1
Sentence: I saw the boy with a telescope

Constituency Parse Using our Model	Dependency Parse Using our Model
	

```
      ROOT
       |
       S
   ___|____________
  |               VP
  |    ___________|________
   |        NP           NP
  |   |     ___|___     ____|______
PRP VBD   DT      NN  IN   DT      NN
 |   |    |       |   |    |       |
 I  saw  the     boy with  a   telescope
```

```
Root
└── ROOT-saw-1
    ├── CASE-I-0
    ├── DOBJ-telescope-6
    │   ├── CASE-with-4
    │   └── DET-a-5
    └── IOBJ-boy-3
        └── DET-the-2
```

Example 2:
Sentence: The gunman sprayed the building with bullets.

Constituency Parse Using our Model	Dependency Parse Using our Model
	

```
                        ROOT
                         |
                         NP
      ___________________|___________
     |                               VP
     |               ________________|____________
     NP             |         NP                 NP
  ___|____          |     ____|_____         ____|_____
 DT       NN       VBN   DT          NN     IN         NNS
 |        |         |    |           |      |           |
The     gunman sprayed the       building with       bullets
```

```
Root
└── ROOT-gunman-1
    ├── DET-The-0
    └── XCOMP-sprayed-2
        ├── DOBJ-bullets-6
        │   └── CASE-with-5
        └── IOBJ-building-4
            └── DET-the-3
```