# Natural Language to Sign Language Sentence

We have used our existing research module, English to PSL Translator, which translates extracted English sentence to equivalent PSL Sentence. We will discuss this module in detail below:

## English to PSL Translator

We will discuss this conversion process briefly below:

### Getting Output from Natural Language Parser

This module uses Stanford's NLP for processing sentences to get the composition of sentence and identification of its constituent elements. The parser gives its output in a bracketed notation. An example sentence and its corresponding parser output is given below:

**Sentence:**   
My dog is chasing a black cat.

**Natural Language Parser Output:**   
(ROOT (S (NP (PRP$ My) (NN dog)) (VP (VBZ is) (VP (VBG chasing) (NP (DT a) (JJ black) (NN cat)))) (. .)))

Stanford's Natural Language Parser uses the grammar tags from Penn Treebank Tag-set for tagging linguistic structure of natural languages.

### Saving Grammar Information in a Tree Data-Structure

To process the grammar information of English Sentences, and translating them into Pakistan Sign Language sentences. The output from Natural Language Parser is stored in a tree data-structure.

For the sentence example given in previous section, the NLP output is

**Natural Language Parser Output:**   
(ROOT (S (NP (PRP$ My) (NN dog)) (VP (VBZ is) (VP (VBG chasing) (NP (DT a) (JJ black) (NN cat)))) (. .)))

The tree data-structure constructed from this output can be viewed as given in following diagram:

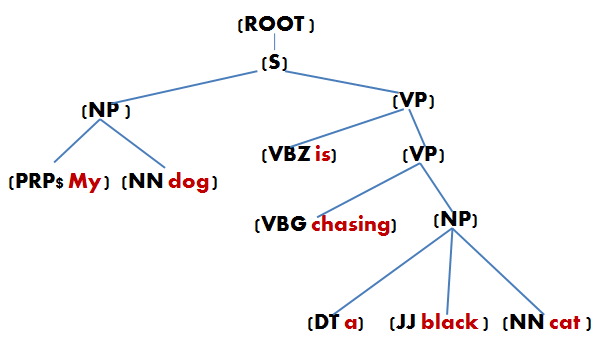


Figure 20: English Tree constructed from NLP Output

### graph.pngGetting Word Dependencies from Typed Dependencies Tool

Another tool called Stanford's typed dependences Tool, gives information about the dependencies between the words of input sentence. This dependency output actually represents a directed graph. Graphical representation of the graph is given in the following diagram:

In this graph, nodes represent the words, and the edges represent the dependencies between the words.

Figure 21: Dependency Graph

### Annotated Tree

The tree data structure constructed from NLP output, is annotated by adding the word dependency information in it.

The following diagrams show the input tree and the output tree after the addition of typed dependencies information.

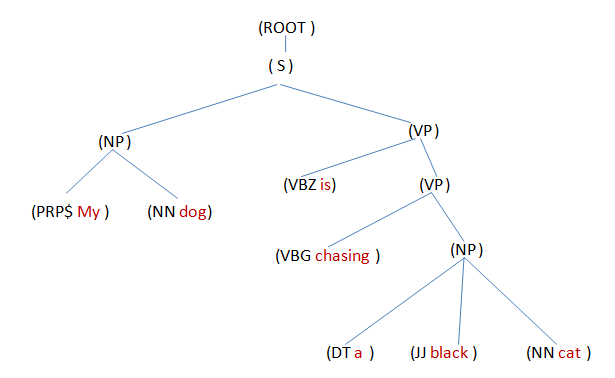


Figure 22: Input Tree

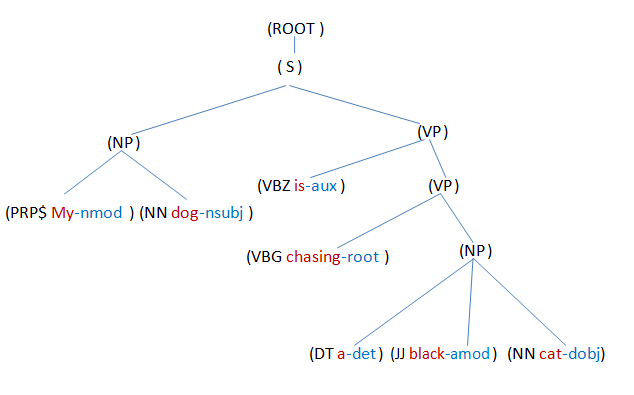


Figure 23: Output Tree (Annotated)

As you can see that our tree data structure now has additional information stored in word-level nodes.

### Renaming of English Tree according to PSL Grammar

In this constructed English language tree, the words are tagged using Treebank Tag-set. These tags are different than the tags (names) in the grammar was developed for Pakistan Sign Language, which contains simplified and easy to understand tag-set.

Therefore, after the identification of sentence type and tense, the Penn tree-bank tags are renamed to the PSL grammar tags using a mapping function. The following diagrams show the input English tree and the renamed English tree.

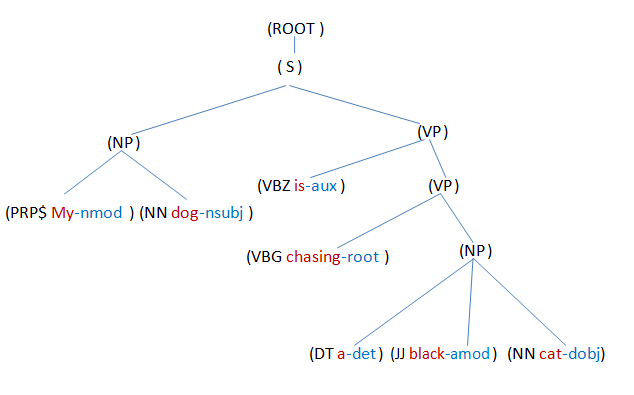


Figure 26: Input English Tree

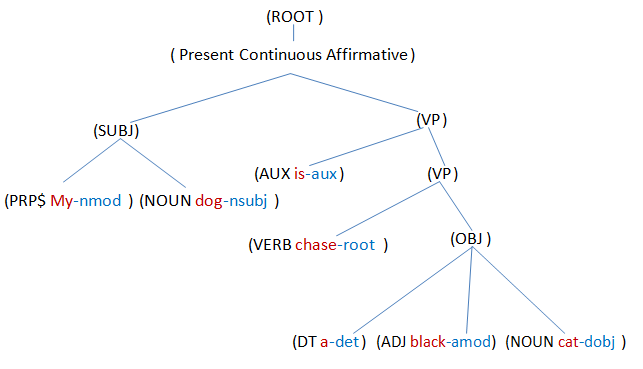


Figure 27: Output renamed English Tree

The tags renamed in this example are given in the following table:

|  |  |
| --- | --- |
| English Tag | PSL Tag |
| S | Present Continuous Affirmative |
| NP | SUBJ |
| PRP$ | PRPS |
| NN | NOUN |
| VBZ | AUX |
| VBG | VERB |
| NP | OBJ |
| JJ | ADJ |

### Construction of PSL Tree

After the renaming of English sentence tree, the tree is parsed from top to bottom (pre-order traversal).

PSL grammar rules are input to the program in form of a text file. This file contains PSL grammar productions written in Backus-Naur form. At each non-terminal (non-leaf) node, the children of that node are compared by the PSL grammar productions for that non-terminal and a best matched production is chosen.

The tree is modified according to the matched PSL production. The modification operations on a non-terminal node are of three types:

1. Addition of a new child node
2. Removal of a child node
3. Re-ordering of children nodes

The new tree thus constructed have all productions according to PSL grammar.

Let us take the example of the sentence used in previous sections:

**Input Sentence:**   
My dog is chasing a black cat.

After performing series of these operations on this sentence's English tree to convert it to an equivalent PSL tree, the output tree can be observed below:

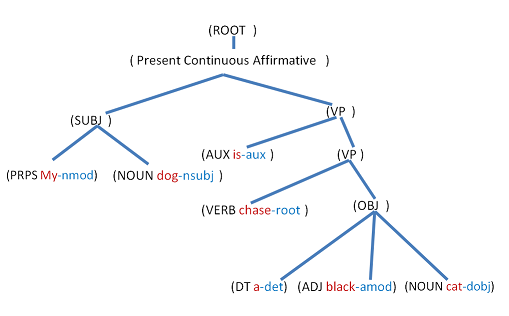


Figure 21: Renamed English Tree

Therefore the final PSL tree will be:

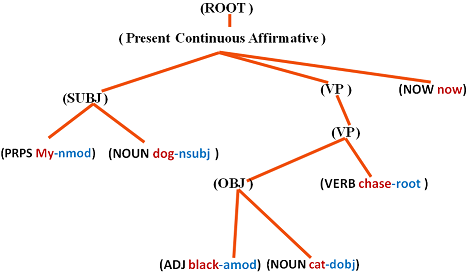


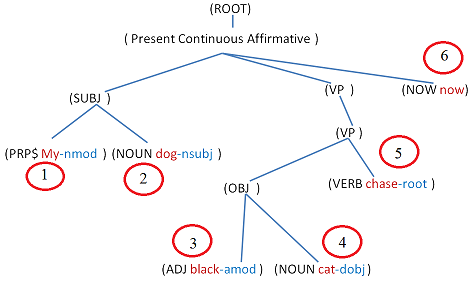
Figure 21: Equivalent PSL Tree

### Translated Sentence

The PSL tree is traversed again for a leaf-only traversal. And the value (word) encountered on each leaf node is concatenated in a resultant string variable.

After the traversal, we get PSL sentence in the resultant string variable.

The following diagram shows the order in which the leaf nodes will be visited.



The resultant PSL sentence will be:

**My dog black cat chase now.**

Likewise, this module takes English sentences as input and return their equivalent PSL Sentence to our user interface.