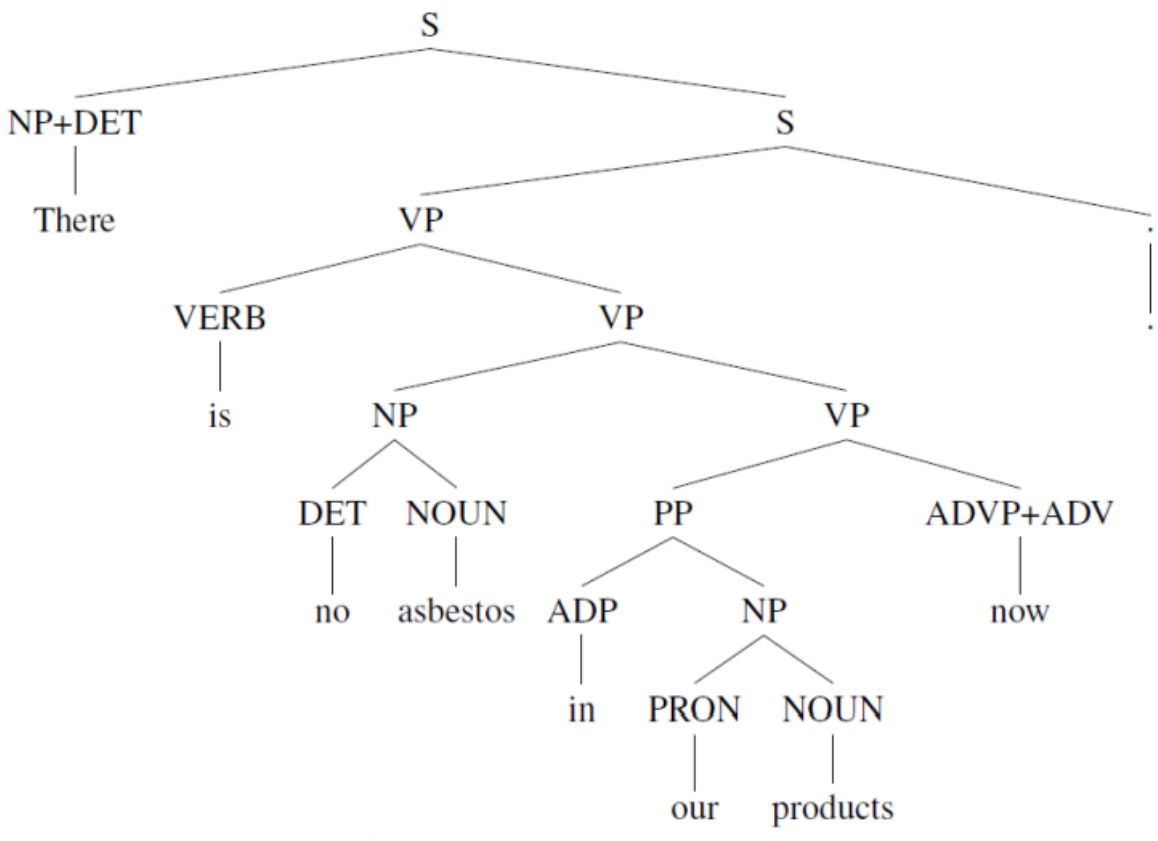
Syntactic natural language parser

1. What is syntactic natural language parser?

   Syntactic natural language parser aims to compute the grammatical structure of sentences. For instance, which groups of words go together (as “phrases”) and which word is the subject or object of a verb. Here is an example. The parser analyzes the sentence in a tree structure. We can see that “our” and “products” are grouped together as a Noun Phrase (NP). Then “in” and “our products” are merged into a preposition phrase (PP). In the end, “There” and other words are grouped as a start symbol (S).



2. How does it work?

   The parser firstly reads a training file, which is a collection of manually annotated sentences. For this project, the training file is Wall street journal corpus. Then it counts the frequency of each grammar rule. The more frequent a rule is, the more likely it will be chosen to build the grammatical structure. The training phase is done.

   Now the parser can work. At first, it reads a sentence in plain text. Then compute and merge the most likely grammatical phrases step by step. At each step, only two phrases can

be merged. The likelihood to merge two phrases depends on the current rule and the previous rules used to build the current phrases.

The parser finally merges the whole sentence as a phrase and outputs the grammatical structure of the sentence.

3. How to run the code?

The code is written in python 2.7. Run “main.py” to train the parser and build the grammatical structure of testing sentences. All the data files are in the “data” directory. “parse_train.dat” is the training file. “parse_dev.dat” is the testing file. “parse_dev.out” is the annotated file. “parse_dev.read” is the more readable format of “parse_dev.out”. It also evaluates the accuracy of the parser in the end. Here is a snapshot of the result. The total F1-score shows the parser’s accuracy is 80.4%.

Type	Total	Precision	Recall	F1-Score
ADVP	1	1.000	1.000	1.000
NP	66	0.658	0.758	0.704
PP	25	0.808	0.840	0.824
S	1	0.000	0.000	0.000
SBAR	1	0.000	0.000	0.000
SBARQ	25	1.000	1.000	1.000
SQ	25	0.960	0.960	0.960
VP	18	0.800	0.222	0.348
WHADJP	2	1.000	1.000	1.000
WHADVP	4	1.000	1.000	1.000
WHNP	21	0.950	0.905	0.927
total	189	0.815	0.794	0.804

4. Here is a simple example

Sentence: What is a nanometer ?

Readable result:

```
[SBARQ,
 [WHNP+PRON, What],
 [SBARQ, [SQ, [VERB, is], [NP, [DET, a], [NOUN, nanometer]]],
[., ?]]]
```

Tree structure: