

# NLP HW5

## 1. Learning a grammar

Q1.a

→ Because there are lots of words that rarely shows in sentences, the sparse data problem

Q1.b

→ Add prime because there are some grammar rules did not show in training sets, so we could generate some new rules using the same properties (such as VP') instead of a particular node (NP-PP-ADVP)

Q1.c

# of binary rules: 226

# of unary rules: 64

# of lexical rules: 309

## 2. CKY Parser

Q2.a

→ Some pre\_terminal could go to several terminals such as NN and VB. Then, count all those possibilities. Thus, the number of strings generated by two grammars is  $2*3*3*2*3+3*3*3=135$ .

Q2.b

→ see toy.parsed

Q2.c

`less test.parsed | grep -c "NONE"`

→ There are 22 sentences parsing failures. Since some words in the test file are not observed in our training set, our grammar.pcfg.bin does not include rules that from preterminal to those new words, and it occurs the parsing failure.

Q2.d

`less test.parsed.new | grep -c "NONE"`

→ There is 0 parsing failure.

Q2.e

→ we used evalb-Copy1.py adding F-1 score calculator in evalb.py.

```
[ $ python evalb-Copy1.py test.parsed test.trees
test.parsed      214 brackets
test.trees       385 brackets
matching         199 brackets
precisions       92.99%
recalls          55.58%
F-1 score        69.58%
```

```
[ $ python evalb-Copy1.py test.parsed.new test.trees
test.parsed.new  371 brackets
test.trees       385 brackets
matching         314 brackets
precisions       84.64%
recalls          96.36%
F-1 score        90.12%
```