Assignment 8

Anurag Dhaipule

Question 1:

The given grammar is:

s = np vp
vp = v np pp
vp = v np
np = n
np = n pp
pp = p np

Probabilities for the productions obtained from training sentences are:

P(s → np vp) = 1
P(vp → v np pp) = 0.4
P(vp → v np) = 0.6
P(np → n) = 0.8
P(np → n np) = 0.2
P(pp → p np) = 1

The two parses for the sentence "Delis serve pizza with relish" are:

Parse 1:
(s (np (n Delis)) (vp (v serve) (np (n pizza)) (pp (p with)(np (n relish)))))
The probability of Parse 1 is:
P(Parse 1) = P(s → np vp) * P(np→n) * P(vp→v np pp) * P(np→n) * P(pp →p np) * P(np → n)
= 1 * 0.8 * 0.4 * 0.8 * 1 * 0.8
P(Parse 1) = **0.2048**

Parse 2 : (s (np (n Delis)) (vp (v serve) (np (n pizza) (pp (p with) (np (n relish))))))
The probability of Parse 2 is:
P(Parse 2) = P(s→np vp) * P(np→n) * P(vp→v np) * P(np→n pp) * P(pp→p np) * P(np → n)
= 1 * 0.8 * 0.6 * 0.2 * 1 * 0.8
P(Parse 2) = **0.0768**
Parse 1 has higher probability and it will be chosen.

Question 2 :

a.
From given training corpus,
Again we calculate it similar to how we did in question 1. Notice that here we are including the head nodes as well in productions.
P(s → np vp | s) = 5/5 = 1
P(np → n pp | np) = 3/15 = 1/5
P(np → n | np) = 12/15 = 4/5
P(pp → p np | pp) = 5/5 = 1
P(vp → v np | vp, like) = 2/2 = 1
P(vp → v np pp | vp, like) = 0/2 = 0
P(vp → v np | vp, serve) = 1/3
P(vp → v np pp | vp, serve) = 2/3

Parse 1:
(s (np (n Delis)) (vp (v serve) (np (n pizza)) (pp (p with)(np (n relish)))))
P(Parse 1)=P(s→np vp | s)*P(np→n | np)*P(vp→v np pp |vp, serve)*P(np →n | np)*P(pp→p np | pp)*P(np→n | np)

= 1 * 4/5 * 2/3 * 4/5 * 1 * 4/5
P (Parse 1) = **0.341**

Parse 2 :
(s (np (n Delis)) (vp (v serve) (np (n pizza) (pp (p with) (np (n relish))))))
P (Parse 2) = P(s → np vp | s) * P(np → n | np) * P(vp → v np | vp, serve) * P(np → n pp | np) * P(pp → p np | pp) * P(np → n | np)
= 1 * 4/5 * 1/3 * 1/5 * 1 * 4/5
P (Parse 2) = **0.0426**

Parse 1 has higher probability and it will be chosen.

b.

Parse 1: (s (np (n Men)) (vp (v like) (np (n pizza)) (pp (p with) (np (n relish)))))
Parse 2: (s (np (n Men)) (vp (v like) (np (n pizza) (pp (p with) (np (n relish))))))

Lexicalized Probability:

P(Parse 1) = P(s → np vp | s) * P(np → n | np) * P(vp → v np pp | vp, like) * P(np → n | np) * P(pp → p np | pp) *P(np → n | np)
= 1 * 4/5 * 0 * 4/5 * 1 * 4/5
P(Parse 1) = **0**

P(parse 2) = P(s → np vp | s) * P(np → n | np) * P(vp → v np | vp, like)*P(np → n pp | np) * P(pp → p np | pp) * P(np → n | np)
= 1* 4/5 * 1 * 1/5 * 1 * 4/5
P(Parse 2) = **0.128**
Parse 2 has higher probability and it will be chosen.

Non-lexicalized Probability:

P(Parse 1) = P(s → np vp) * P(np → n) * P(vp → v np pp) * P(np → n) * P(pp → p np) * P(np → n)
= 1 * 0.8 * 0.4 * 0.8 * 1 * 0.8
P(Parse 1) = 0.2048

P(Parse 2) = P(s → np vp) * P(np → n) * P(vp → v np) * P(np → n pp) * P(pp → p np) * P(np → n)
= 1 * 0.8 * 0.6 * 0.2 * 1 * 0.8
P(Parse 2) = 0.0768
Parse 2 has higher probability and it will be chosen.
Notice that with lexicalized probability we ended up choosing the right parse but with non-lexicalized we chose wrong parse. Hence, lexicalized is better.