Comp550 Natural Language Processing

Assignment 2

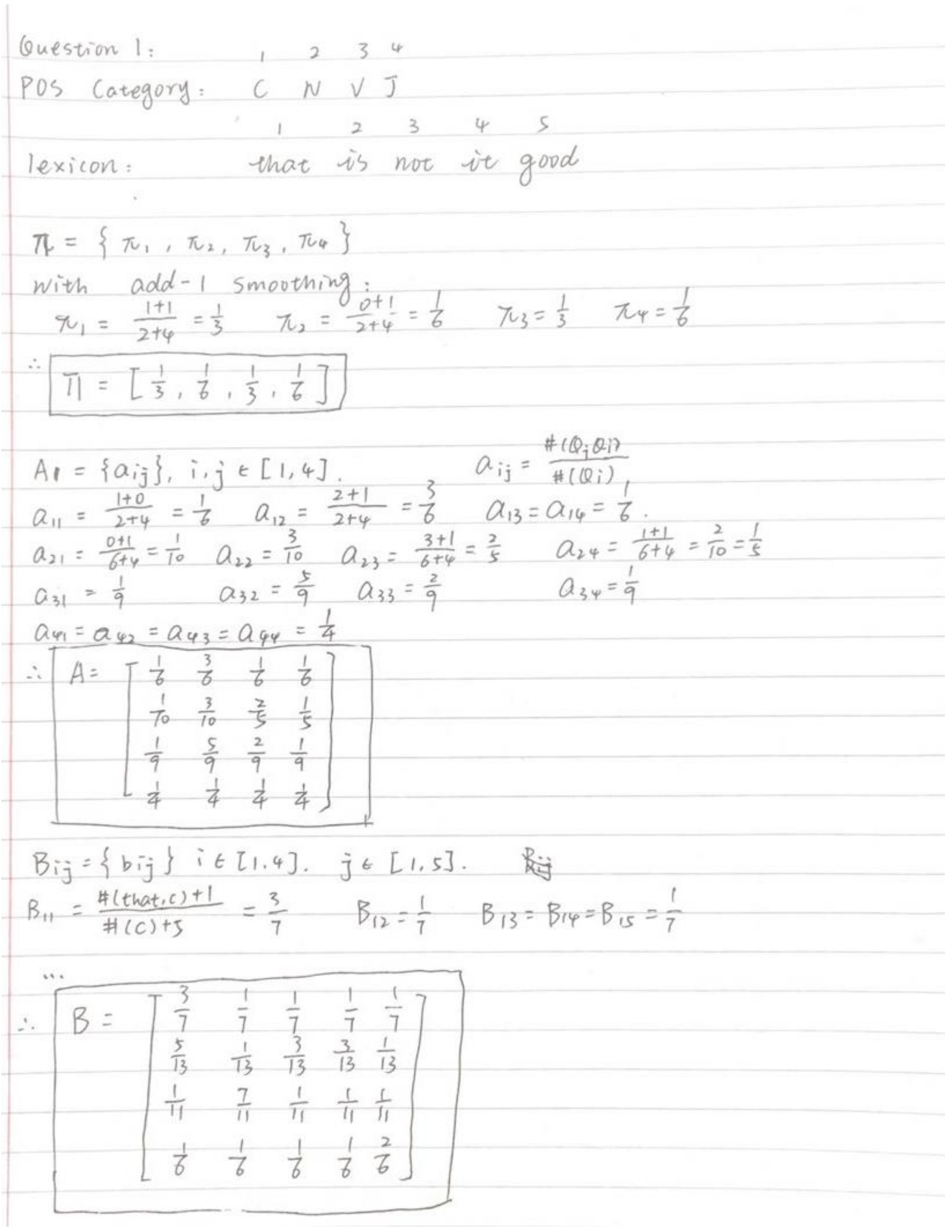
Yiran Mao

Mcgill ID: 260850827

Date: 2018-10-10

Question 1

(1)For the first three parts of this question (Train a first-order HMM part-of-speech tagger; Find the MAP estimate of the parameters of the model; run the Viterbi algorithm to obtain a POS tagging for the sentence “That is good.”), I calculated the result by hand.



Question 1:

	1	2	3	4	
POS Category:	C	N	V	J	
	1	2	3	4	5
lexicon:	that	is	not	it	good

$\pi = \{\pi_1, \pi_2, \pi_3, \pi_4\}$

with add-1 smoothing:

$\pi_1 = \frac{1+1}{2+4} = \frac{1}{3}$ $\quad \pi_2 = \frac{0+1}{2+4} = \frac{1}{6}$ $\quad \pi_3 = \frac{1}{3}$ $\quad \pi_4 = \frac{1}{6}$

$$\therefore \boxed{\pi = \left[\frac{1}{3}, \frac{1}{6}, \frac{1}{3}, \frac{1}{6}\right]}$$

$A = \{a_{ij}\},\ i,j \in [1,4].$ $\quad a_{ij} = \frac{\#(Q_j Q_i)}{\#(Q_i)},$

$a_{11} = \frac{1+0}{2+4} = \frac{1}{6}$ $\quad a_{12} = \frac{2+1}{2+4} = \frac{3}{6}$ $\quad a_{13} = a_{14} = \frac{1}{6}.$

$a_{21} = \frac{0+1}{6+4} = \frac{1}{10}$ $\quad a_{22} = \frac{3}{10}$ $\quad a_{23} = \frac{3+1}{6+4} = \frac{2}{5}$ $\quad a_{24} = \frac{1+1}{6+4} = \frac{2}{10} = \frac{1}{5}$

$a_{31} = \frac{1}{9}$ $\quad a_{32} = \frac{5}{9}$ $\quad a_{33} = \frac{2}{9}$ $\quad a_{34} = \frac{1}{9}$

$a_{41} = a_{42} = a_{43} = a_{44} = \frac{1}{4}$

$$\therefore \boxed{A = \begin{bmatrix} \frac{1}{6} & \frac{3}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{10} & \frac{3}{10} & \frac{2}{5} & \frac{1}{5} \\ \frac{1}{9} & \frac{5}{9} & \frac{2}{9} & \frac{1}{9} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}}$$

$B_{ij} = \{b_{ij}\}$ $i \in [1,4].$ $j \in [1,5].$ B_{ij}

$B_{11} = \frac{\#(that, C)+1}{\#(C)+5} = \frac{3}{7}$ $\quad B_{12} = \frac{1}{7}$ $\quad B_{13} = B_{14} = B_{15} = \frac{1}{7}$

...

$$\therefore \boxed{B = \begin{bmatrix} \frac{3}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{5}{13} & \frac{1}{13} & \frac{3}{13} & \frac{3}{13} & \frac{1}{13} \\ \frac{1}{11} & \frac{7}{11} & \frac{1}{11} & \frac{1}{11} & \frac{1}{11} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{2}{6} \end{bmatrix}}$$

1 2 5

Viterbi algorithm for sentence: That is good.

$\delta_i(t) = \max_{Q_1:t-1} P(Q_1:t-1,\ Q_1:t,\ Q_t = i \mid \theta)$

$\delta_i(1) = \pi_i b_i(O_1) = \pi_i b_{i1} \quad \delta_1(1) = \frac{1}{3} \times b_{11} = \frac{1}{3} \times \frac{3}{7} = \frac{1}{7}$

$\delta_2(1) = \frac{1}{6} \times \frac{5}{13} = \frac{5}{78} \quad \delta_3(1) = \frac{1}{3} \times \frac{1}{11} = \frac{1}{33} \quad \delta_4(1) = \frac{1}{6} \times \frac{1}{6} = \frac{1}{36}$

$\delta_1(2) = \max_i \delta_{i1} a_{i1} b_{12}$

$= \max \left[\frac{1}{7} \times \frac{1}{6} \times \frac{1}{7},\ \frac{5}{78} \times \frac{1}{10} \times \frac{1}{7},\ \frac{1}{33} \times \frac{1}{9} \times \frac{1}{7},\ \frac{1}{36} \times \frac{1}{4} \times \frac{1}{7} \right]$

$= \max \left[\bigcirc\!\!\!\!\frac{1}{294} \quad \frac{1}{1092} \quad \frac{1}{2079} \quad \frac{1}{1008} \right]$

$= \frac{1}{294}$ and $Q_1 = C$.

$\delta_2(2) = \max \delta_{i1} a_{i2} b_{22}$

$= \max \left[\frac{1}{7} \times \frac{3}{6} \times \frac{1}{13},\ \frac{5}{78} \times \frac{3}{10} \times \frac{1}{13},\ \frac{1}{33} \times \frac{5}{9} \times \frac{1}{13},\ \frac{1}{36} \times \frac{1}{4} \times \frac{1}{13} \right]$

$= \max \left[\bigcirc\!\!\!\!\frac{1}{182} \quad \frac{1}{676} \quad \frac{1}{3861} \quad \frac{1}{1872} \right)$

$= \frac{1}{182}$ and $Q_1 = C$.

$\delta_3(2) = \max \left[\frac{1}{7} \times \frac{1}{6} \times \frac{7}{11},\ \frac{5}{78} \times \frac{2}{5} \times \frac{2}{11},\ \frac{1}{33} \times \frac{2}{9} \times \frac{7}{11},\ \frac{1}{36} \times \frac{1}{4} \times \frac{7}{11} \right]$

$= \left[\frac{1}{66},\ \bigcirc\!\!\!\!\frac{14}{858},\ \frac{14}{3267},\ \frac{7}{1584} \right]$ and $Q_1 = N$.

$\delta_4(2) = \max \left[\frac{1}{7} \times \frac{1}{6} \times \frac{1}{6} \quad \frac{8}{75} \times \frac{1}{3} \times \frac{1}{6} \quad \frac{1}{33} \times \frac{1}{9} \times \frac{1}{6} \quad \frac{1}{36} \times \frac{1}{4} \times \frac{1}{6} \right]$

$= \max \left[\bigcirc\!\!\!\!\frac{1}{252} \quad \frac{4}{1125} \quad \frac{1}{1782} \quad \frac{1}{864} \right]$

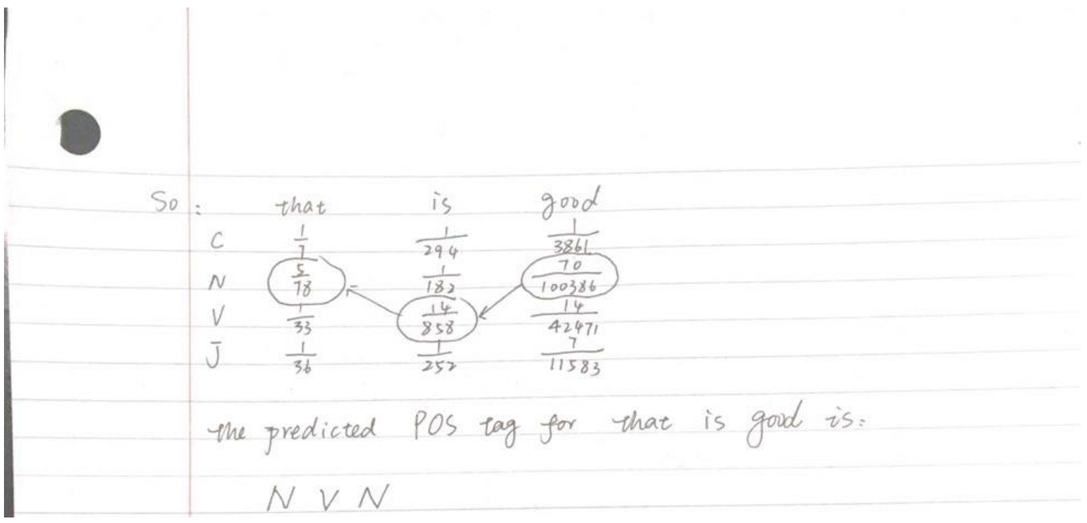
$= \frac{1}{252}$ and $Q_1 = C$.

Then ... $\delta_1(3) = \frac{1}{3861} \quad Q_2 = V \left(\max \left[\frac{1}{294} \times \frac{1}{6} \times \frac{1}{7} \quad \frac{1}{182} \times \frac{1}{10} \times \frac{1}{7} \quad \frac{14}{858} \times \frac{1}{9} \times \frac{1}{7} \quad \frac{1}{252} \times \frac{1}{4} \times \frac{1}{7} \right) \right.$

$\delta_2(3) = \frac{70}{100386} \quad Q_2 = V \left(\max \left[\frac{1}{294} \times \frac{3}{6} \times \frac{1}{13} \quad \frac{1}{182} \times \frac{3}{10} \times \frac{1}{13} \quad \frac{14}{858} \times \frac{5}{9} \times \frac{1}{13} \quad \frac{1}{252} \times \frac{1}{4} \times \frac{1}{13} \right. \right.$

$\delta_3(3) = \frac{14}{42471} \quad Q_2 = V. \left(\max \left[\frac{1}{296} \times \frac{1}{6} \times \frac{1}{11} \quad \frac{1}{182} \times \frac{2}{5} \times \frac{1}{11} \quad \frac{14}{858} \times \frac{2}{9} \times \frac{1}{11} \quad \frac{1}{252} \times \frac{1}{4} \times \frac{1}{11} \right] \right.$

$\delta_4(3) = \frac{7}{11583} \quad Q_2 = V \left(\max \left[\frac{1}{294} \times \frac{1}{6} \times \frac{2}{6} \quad \frac{1}{182} \times \frac{2}{5} \times \frac{2}{6} \quad \frac{14}{858} \times \frac{1}{9} \times \frac{2}{6} \quad \frac{1}{252} \times \frac{1}{4} \times \frac{2}{6} \right] \right.$



So:

	that	is	good
C	1/7	1/294	1/3861
N	5/78	1/182	70/100386
V	1/33	14/858	14/42471
J	1/36	1/252	7/11583

the predicted POS tag for that is good is:

N V N

The result is:

	that	is	good
C	1/7	1/294	1/3861
N	5/78	1/182	70/100386
V	1/33	14/858	14/42471
J	1/36	1/252	7/11583

The predicted POS tag for “that is good ” is NVN.

(2)Then for the last part of the question(Update your previously trained model with a semi-supervised method involving Expectation Maximization over the additional sentences such that it can now account for the word “bad".), I wrote a matlab code(EM.m) to calculate(I will provided the code EM.m).

Firstly, I re-smooth the model(add word “bad”):

```
pi=[1/3 1/6 1/3 1/6];

A = [1/6 3/6 1/6 1/6;
     1/10 3/10 2/5 1/5;
     1/9 5/9 2/9 1/9;
     1/4 1/4 1/4 1/4];

B = [3/8 1/8 1/8 1/8 1/8 1/8;
    5/14 1/14 3/14 3/14 1/14 1/14;
    1/12 7/12 1/12 1/12 1/12 1/12;
    1/7 1/7 1/7 1/7 2/7 1/7];
```

Secondly calculated the forward and backward algorithm for all 7 sentences:

$\alpha_i(t)$is $P(\boldsymbol{O}_{1:t}, Q_t = i|\theta)$

$$\beta_i(t) = P(\boldsymbol{O}_{t+1:T}|Q_t = i, \theta)$$

Then calculated : (for all 7 sentences)

$$\begin{aligned}\gamma_i(t) &= P(Q_t = i|\boldsymbol{O}, \theta^k) \\ &= \frac{P(Q_t = i, \boldsymbol{O}|\theta^k)}{P(\boldsymbol{O}|\theta^k)} \\ &= \frac{\alpha_i(t)\beta_i(t)}{P(\boldsymbol{O}|\theta^k)}\end{aligned}$$

and

$$\begin{aligned}\xi_{ij}(t) &= P(Q_t = i, Q_{t+1} = j|\boldsymbol{O}, \theta^k) \\ &= \frac{P(Q_t = i, Q_{t+1} = j, \boldsymbol{O}|\theta^k)}{P(\boldsymbol{O}|\theta^k)} \\ &= \frac{\alpha_i(t)a_{ij}b_j(O_{t+1})\beta_j(t+1)}{P(\boldsymbol{O}|\theta^k)} \\ & \quad \nearrow \quad \nearrow \quad \uparrow \quad \uparrow\end{aligned}$$

Finally, calculate the new pi, A, B(sum up expected counts over all 7 sentences.)

$$\pi_i^{k+1} = \gamma_i(1)$$

$$a_{ij}^{k+1} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

$$b_i^{k+1}(w_k) = \frac{\sum_{t=1}^{T} \gamma_i(t)|_{O_t=w_k}}{\sum_{t=1}^{T} \gamma_i(t)}$$

Then I got the result of new model:

A:

0.1480	0.4908	0.2134	0.1478
0.0866	0.2402	0.4624	0.2108
0.0975	0.5819	0.2077	0.1129
0.2174	0.2199	0.3051	0.2576

B:

0.4906	0.1939	0.0552	0.0693	0.0865	0.1045
0.3627	0.0946	0.2226	0.1957	0.0786	0.0458
0.0762	0.7087	0.0361	0.0487	0.0693	0.0610
0.1604	0.2113	0.0908	0.1069	0.3010	0.1297

pi:

0.3453	0.1441	0.4005	0.1101

The probability of the sentence “that is bad” is 47/68834. (use Viterbi algorithm. Calculate in Viterbi.m)

Question 2

My CFG is in the file “plaintext.txt”.

Example of Accept:

Nous regardons la télévision.

going bottom up the grammar will recognize the following constituents: PR-1Pl V-1Pl DT-Sg-F N-Sg-F. Then we can see from my CFG, PR-1Pl will be recognized as NP-1Pl; N-Sg-F will be recognized as NA-Sg-F; DT-Sg-F and NA-Sg-F will combine to form NP-3Sg-F, which will be recognized as NP. V-1Pl and NP will combine to form VP-1Pl. Finally, NP-1Pl and VP-1Pl will combine to form S.

Il la regarde.

going bottom up the grammar will recognize the following constituents: PR-3Sg-M PR-do V-3Sg-M. PR-3Sg-M will be recognized as NP-3Sg-M, PR-do and V-3Sg-M will combine to form VP-3Sg-M. Then NP-3Sg-M and VP-3Sg-M will finally combine to form S.

Example of Reject:

*Je mangent le poisson.

going bottom up the grammar will recognize the following constituents:

PR-1Sg V-3PL-M|V-3PL-F DT-Sg-M N-Sg-M. Then PR-1Sg will be recognized as NP-1Sg, N-Sg-M will be recognized as NA-Sg-M , DT-Sg-M NA-Sg-M will combine to form NP-3Sg-M, which will be recognized as NP. Then V-3PL-M|V-3PL-F and NP will combine to form VP-3PL-M|VP-3PL-F. However NP-1Sg and VP-3PL-M|VP-3PL-F cannot form S(we can’t find such combination in the CFG).

*Les noirs chats mangent le poisson.

Noirs and chats will be recognized as A1-Pl-M N-Pl-M, which cannot form NA-PL-M(in my CFG it should be N-Pl-M A1-Pl-M). So it’s wrong.

*La poisson mangent les chats.

La poisson will be recognized as DT-Sg-F N-Sg-M , they cannot form a NP. So it will be rejected by my CFG.

*Je mange les.

Manage les will be recognized as V-1Sg PR-do. However V-1Sg PR-do cannot combine to form VP-1Sg (it should be PR-do V-1Sg). So it is wrong.

1.What are some advantages of modelling French grammar with a CFG, compared to using an FSA?

There are too many rules involved in French grammar, trying to model it with FSA will become confusing because each node has multiple branches from it. So build a FSA for this model is difficult and complicated. There are also many irregularities, so the FSA will become huge and complex.

2. What are some disadvantages of modelling French grammar with a CFG?

Because each rule has many possible outputs, modeling French grammar using CFG can result in slower parsing. The grammar must undergo many different parsions before it reaches the correct parsing.

3. What are some aspects of French grammar that your CFG does not handle?

My CFG does not handle the indirect object of verb phrases, multiple adjectives, tenses of verbs and so on. In addition, I do not account for the adverbs and prepositions, as well as any proper nouns that take articles.

Question3

In this paper the authors mainly want to use some annotation methods to show that parsing performance that can be achieved by an unlexicalized PCFG is far higher than community wisdom has thought possible.

At first, they explained two deficiencies of the raw n-ary treebank grammar: 1) the category symbols are too coarse to adequately render the expansions independent of the context. 2)probabilities overestimated and underestimated. Then they gave the solutions of above problems: parent annotation and RHS maokovization. In this way, they proposed the vertical and horizontal markovization(only the past v vertical ancestors and the previous h horizontal ancestors matter). they found that v=3 and h<=2 is the best entry. However, it has a large number of states and does not tolerate further splitting well, so they chose the ***baseline*** with v<=2 and h<=2. Then they tried several annotations. ***Firstly*** the UNARY-INTERNAL (marks any nonterminal node which has only one child) and UNARY-EXTERNAL(marks nodes which has no siblings with ^U). They are from the categories: internal and external annotation. external annotation indicates features of the external environment of a node which influences the internal expansion, while internal annotation marks a distinctive aspect of the otherwise hidden internal contents of a node which influence the external distribution. ***Secondly*** they used the method of tag splitting, for example UNARY-DT showed that the determiners which occur alone are distinguished from those which occur with other nominal material. ***Then*** they added some annotations already in the treebank such as TMP-NP and GAPPED-S. ***Also***, some head annotations are considered such as POSS-NP and SPLIT-VP. ***Finally***, BASE-NP, DOMINATES-V and RIGHT-REG-NP are defined to model depth and distance features. They used sections 2-21 of the WSJ section of the Penn treebank as training set, section 22 as a development set, and section 23 as a test set. Their parser really gained a high performance(87.04 at F1).

Lexicalized means using **lexical class words(open class)** to annotate in the grammar parser, such like annotate head word for every node. However, in unlexicalized grammar parser, lexical class words are not part of grammatical structure. It only uses **function words(closed class)** for annotation but never uses any lexical class words to provide either monolexical or bilexical probabilities.

From this paper, I really learned much knowledge: the differences between lexicalized and unlexicalized parsing, the vertical and horizontal markovization, the internal and external annotations, and many other methods of annotations. From the class we learned that PCFG has problem of ambiguity and we need to define head word for every rule to solve this problem. But in this paper, I learned that maybe lexicalization is not as important as we thought before.

Three questions: Can the methods of annotation mentioned in this paper increase the performance of a lexicalized parsing? If there's any other annotation methods to further improve the performance of the unlexicalized parsing? What about the parsing speed of the unlexicalized parser mentioned in the paper?