Comp 550 Assignment 2

Jonathan Pearce, 260672004

October 17, 2018

Problem 1a.

Initial Probabilities $\pi_i = Pr(Q_1 = i)$			
C	N	V	J
$\frac{3}{8}$	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

Transition Probabilities $a_{ij} = Pr(Q_{t+1} = j \mid Q_t = i)$				
	C	N	V	J
C	$\frac{1}{6}$	$\frac{3}{6}$	$\frac{1}{6}$	$\frac{1}{6}$
N	$\frac{1}{10}$	$\frac{3}{10}$	$\frac{4}{10}$	$\frac{2}{10}$
V	$\frac{1}{9}$	$\frac{5}{9}$	$\frac{2}{9}$	$\frac{1}{9}$
J	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

Emission Probabilities $b_{ik} = Pr(O_t = k \mid Q_t = i)$					
	that	is	it	not	good
C	$\frac{3}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$
N	$\frac{5}{13}$	$\frac{1}{13}$	$\frac{3}{13}$	$\frac{3}{13}$	$\frac{1}{13}$
V	$\frac{1}{11}$	$\frac{7}{11}$	$\frac{1}{11}$	$\frac{1}{11}$	$\frac{1}{11}$
J	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{2}{6}$

Problem 1b. Using the Viterbi Algorithm, First Column:

$$\delta_1(1) = \pi_1 \cdot b_{11} = \frac{3}{8} \cdot \frac{3}{7} = \frac{9}{56}$$

$$\delta_2(1) = \pi_1 \cdot b_{21} = \frac{1}{8} \cdot \frac{5}{13} = \frac{5}{104}$$

$$\delta_3(1) = \pi_1 \cdot b_{31} = \frac{3}{8} \cdot \frac{1}{11} = \frac{3}{88}$$

$$\delta_4(1) = \pi_1 \cdot b_{41} = \frac{1}{8} \cdot \frac{1}{6} = \frac{1}{48}$$

Second Column:

$$\delta_1(2) = b_{12} \cdot \max_i(\delta_i(1) \cdot a_{i1})$$

$$= \frac{1}{7} \cdot \max(\frac{9}{56} \cdot \frac{1}{6}, \frac{5}{104} \cdot \frac{1}{10}, \frac{3}{88} \cdot \frac{1}{9}, \frac{1}{48} \cdot \frac{1}{4})$$

$$= 0.003826$$

(backpointer: C)

$$\delta_2(2) = b_{22} \cdot \max_i(\delta_i(1) \cdot a_{i2})$$

$$= \frac{1}{13} \cdot \max(\frac{9}{56} \cdot \frac{3}{6}, \frac{5}{104} \cdot \frac{3}{10}, \frac{3}{88} \cdot \frac{5}{9}, \frac{1}{48} \cdot \frac{1}{4})$$

$$= 0.006181$$

(backpointer: C)

$$\delta_3(2) = b_{32} \cdot \max_i(\delta_i(1) \cdot a_{i3})$$

$$= \frac{7}{11} \cdot \max(\frac{9}{56} \cdot \frac{1}{6}, \frac{5}{104} \cdot \frac{4}{10}, \frac{3}{88} \cdot \frac{2}{9}, \frac{1}{48} \cdot \frac{1}{4})$$

$$= 0.01704$$

(backpointer: C)

$$\delta_4(2) = b_{42} \cdot \max_i(\delta_i(1) \cdot a_{i4})$$

$$= \frac{1}{6} \cdot \max(\frac{9}{56} \cdot \frac{1}{6}, \frac{5}{104} \cdot \frac{2}{10}, \frac{3}{88} \cdot \frac{1}{9}, \frac{1}{48} \cdot \frac{1}{4})$$

$$= 0.004464$$

(backpointer: C)

Third Column:

$$\delta_1(3) = b_{15} \cdot \max_i(\delta_i(2) \cdot a_{i1})$$

$$= \frac{1}{7} \cdot \max(0.003826 \cdot \frac{1}{6}, 0.006181 \cdot \frac{1}{10}, 0.01704 \cdot \frac{1}{9}, 0.004464 \cdot \frac{1}{4})$$
$$= 0.0002705$$
(backpointer: V)

$$\delta_2(3) = b_{25} \cdot \max_i(\delta_i(2) \cdot a_{i2})$$
$$= \frac{1}{13} \cdot \max(0.003826 \cdot \frac{3}{6}, 0.006181 \cdot \frac{3}{10}, 0.01704 \cdot \frac{5}{9}, 0.004464 \cdot \frac{1}{4})$$
$$= 0.0007282$$
(backpointer: V)

$$\delta_3(3) = b_{35} \cdot \max_i(\delta_i(2) \cdot a_{i3})$$
$$= \frac{1}{11} \cdot \max(0.003826 \cdot \frac{1}{6}, 0.006181 \cdot \frac{4}{10}, 0.01704 \cdot \frac{2}{9}, 0.004464 \cdot \frac{1}{4})$$
$$= 0.0003442$$
(backpointer: V)

$$\delta_4(3) = b_{45} \cdot \max_i(\delta_i(2) \cdot a_{i4})$$
$$= \frac{2}{6} \cdot \max(0.003826 \cdot \frac{1}{6}, 0.006181 \cdot \frac{2}{10}, 0.01704 \cdot \frac{1}{9}, 0.004464 \cdot \frac{1}{4})$$
$$= 0.0006311$$
(backpointer: V)

Therefore,

$\delta_i(j)$ with backpointers)			
	That	is	good
C	$\frac{9}{56}$	0.003826 (C)	0.0002705 (V)
N	$\frac{5}{104}$	0.006181 (C)	0.0007282 (V)
V	$\frac{3}{88}$	0.01704 (C)	0.0003442 (V)
J	$\frac{1}{48}$	0.004464 (C)	0.0006311 (V)

Finally,

$$\max_i \delta_i(3) = \delta_2(3) = 0.0007282$$

Therefore N is the most likely tag for the word 'good' and following the backpointers we find that the most likely sequence of tags for the test sentence is,

That/C is/V good/N.

Problem 1c. First we have to incorporate 'bad' into our emission probabilities. Since our labelled sentences did not contain 'bad' we will assume the probability of emitting 'bad' given a tag is 0. However, now in our emission probability table we have an extra category which requires use to redo add-1 smoothing. After this process, this is our new emission table:

Emission Probabilities $b_{ik} = Pr(O_t = k \mid Q_t = i)$						
	that	is	it	not	good	bad
C	$\frac{3}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
N	$\frac{5}{14}$	$\frac{1}{14}$	$\frac{3}{14}$	$\frac{3}{14}$	$\frac{1}{14}$	$\frac{1}{14}$
V	$\frac{1}{12}$	$\frac{7}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$
J	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{2}{7}$	$\frac{1}{7}$

I did my calculations for expectation maximization in an excel file which is attached in my assignment submission (final tables are highlighted green). After running expectation maximization on the 2 new unlabelled sentences these are the new probability tables:

Initial Probabilities $\pi_i^{k+1} = Pr(Q_1 = i)$			
C	N	V	J
0.2745	0.07606	0.5254	0.1241

Transition Probabilities $a_{ij}^{k+1} = Pr(Q_{t+1} = j \mid Q_t = i)$				
	C	N	V	J
C	0.1403	0.3540	0.3188	0.1809
N	0.1072	0.2156	0.33792	0.3392
V	0.08670	0.5993	0.1975	0.1165
J	0.2086	0.1793	0.3099	0.3021

Emission Probabilities $b_{ik}^{k+1} = Pr(O_t = k \mid Q_t = i)$						
	that	is	it	not	good	bad
C	0.3750	0.1452	0.05254	0.05403	0.07132	0.3019
N	0.3571	0.05119	0.2107	0.1997	0.06261	0.1187
V	0.08333	0.5315	0.03849	0.04113	0.08467	0.2209
J	0.1429	0.09951	0.07904	0.09520	0.2797	0.3037

NOTE: After running EM for the 2 unlabelled sentences the emission probability table contained 0 values for any cell in the 'that' column (this table is in the excel file) because 'that' was not a word in either of the sentences. Since these two new sentences provided us with no information about the emission of 'that' given a tag I choose to copy the values from the 'that' column from the original emission probability table into the updated emission table. After this I normalized the other cell values (every column except 'that') so that each row summed to 1 and formed a probability distribution. This solution ensures that unseen sentences such as 'That is bad' will not be given a 0 probability estimate

Problem 2. Accepting sentences using my grammar:

Il regarde la télévision.

From our grammar we have, PR-3Sg -> Il, V-3Sg -> regarde, DT-Fem -> la and N-SgFem -> télévision. Therefore this sentence of non terminals can be converted to a sentence of terminals and then working backwards with our rules we get:

PR-3Sg V-3Sg DT-Fem N-SgFem
↔ NP-3Sg V-3Sg DT-Fem NP-Fem
↔ NP-3Sg V-3Sg NP
↔ NP-3Sg VP-3Sg
↔ S

Le beau chat

From our grammar we have, DT-Masc -> Le, A-PrecedeSgMasc -> beau and N-SgMasc -> chat. Therefore this sentence of non terminals can be converted to a sentence of terminals and then working backwards with our rules we get:

DT-Masc A-PrecedeSgMasc N-SgMasc
↔ DT-Masc NP-Masc
↔ NP
↔ S

Rejecting sentences using my grammar:

*Je mangent le poisson.

From our grammar we have, PR-1Sg -> Je, V-3Pl -> mangent, DT-Masc -> le and N-SgMasc -> poisson. Therefore this sentence of non terminals can be converted to a sentence of terminals and then working backwards with our rules we get:

PR-1Sg V-3Pl DT-Masc N-SgMasc
↔ NP-1Sg V-3Pl DT-Masc NP-Masc
↔ NP-1Sg V-3Pl NP
↔ NP-1Sg VP-3Pl
↔ NULL

*Je mange les.

From our grammar we have, PR-1Sg -> Je, V-1Sg -> mange and DT-Pl -> les. Therefore this sentence of non terminals can be converted to a sentence of terminals and then working backwards with our rules we get:

PR-1Sg V-1Sg DT-Pl
↔ NP-1Sg V-1Sg NULL
↔ NP-1Sg NULL
↔ NULL

1. The French language contains many subtle intricacies some of which we modelled in our CFG such as adjectives that can follow or precede the noun they modify and direct object pronouns. The CFG is quite efficient at handling these special cases and can be easily adapted to fit other aspects of the French language, in most cases a few terminal rules and modifying or creating a couple of non terminal rules is all that is required. Modelling these subtleties in an FSA is much more complicated, as each node could have quite a few branches coming into it or leaving it, further adding new rules or information about a language into a FSA could require it to need serious expansion or even possibly require that it be rebuilt with a different structure.

2. The downside of the CFG is the time it takes to parse. Most parses will be inefficient since each rule has many possible outputs leading to a significant number of total parses to check through before finding the correct one

3. We did not fully model the direct object pronouns, specifically whether a verb takes a direct object or not depends on whether the specific verb is transitive or intransitive, these verb properties were not considered and thus our modelling of direct object pronouns in incomplete. Our CFG also only considers verbs in the present tense, past and future are ignored.

Problem 3a. This paper by Klein and Manning serves as a demonstration that unlexicalized PCFGs are a more powerful tool in accurate parsing then previously thought, when compared to the performance of lexicalized PCFG models. The paper is not an argument against lexicalized parsing, in fact Klein and Manning show that their unlexicalized PCFG is more accurate than early lexicalized models, however state of the art lexicalized PCFGs still come out on top. The motivation for this work came from the fact that unlexicalized PCFGs have an incredible number of benefits in comparison to their lexicalized counterparts. In fact, unlexicalized PCFGs are easier to interpret, the grammar representation is more compact, parsing algorithms such as CKY have lower asymptotic complexity and they are much simpler to build and optimize. Clearly if an unlexicalized PCFG could be developed with reasonable parsing accuracy, it could have significant benefits to areas of research where the best lexicalized models are too slow.

All data for this work was taken from the Wall Street Journal section of the Penn treebank. The unlexicalized PCFG was trained on sections 2 to 21. Next the first 20 files of section 22 was used as a development set. Finally section 23 was used as a test set for the final unlexicalized model. Throughout the paper F1 score is the evaluation metric used for model evaluation and comparison with lexicalized PCFGs.

Klien and Manning also take time in the paper to explicitly define the difference between unlexicalized and lexicalized PCFGs. The distinction comes in the treatment of words in the open/lexical word class. Lexicalized PCFGs sub-categorize words in closed classes (i.e. function words) to better represent important distinctions, and, features that are commonly expressed by closed class words are annotated onto phrasal nodes. Further words in open classes (i.e. content words) can be processed and used to provide monolexical and bilexical probabilities. In comparison unlexicalized PCFGs only make use of the sub-categorization of function words, and make no use of content words.

Problem 3b. On a broad scale this paper showed me the advantages of unlexicalized PCFGs in terms of complexity both in space and parsing time, and demonstrated that contrary to previous opinion is it possible to develop an unlexicalized PCFG that is comparable in parsing accuracy with state of the art lexicalized PCFGs. More specifically the paper showed me multiple annotation methods that were used to split the symbol space in order to achieve accuracy (F1 score) gains, this included markovization (horizontal and vertical) which involves keeping track of the vertical and horizontal ancestors of a node to a depth defined by parameters h and v. Further methods included comparing internal and external annotation, tag splitting and head annotation. The depth to which Klein and Manning went to annoatating the symbol space demonstrated the underlying complexity of PCFGs.

The advantages of this method is that Klein and Manning have developed a PCFG that can be parsed with accuracy near the state of the art methods, while keeping the PCFG in the unlexicalized domain and therefore ensuring that the grammar representation is more compact, parsing algorithms such as CKY have lower asymptotic complexity. In general Klein and Manning have proven that unlexicalized PCFGs are powerful tools, and many of their steps in improving their model could be translated over to lexicalized PCFGs.

One of the limitations of this work (and in general for unlexicalized PCFG) is the overall accuracy of the parsing. State of the art lexicalized PCFGs still outperform Klein and Manning's model and thus the practically of exclusively using an unlexicalized PCFGs is not

apparent. Perhaps combining lexicalized and unlexicalized PCFGs could help make use of Klein and Manning's work in real examples.

The F1 score of Klein and Manning's final model on the test set was 86.3%, where as the basic unannotated unlexicalized model had an F1 score of 71.3%, this demonstrates that annotations and sub categorizations are critical in developing models that can accurately describe the syntax of natural language as discussed in class during lectures 10 and 11.

Problem 3c.

1. Were there any annotations techniques that proved to be unsuccessful at improving parsing accuracy?

2. Are there practical examples where a lexicalized PCFG would be too slow or infeasible and thus this unlexicalized model would be required to improve parsing time and space efficiency?

3. Could further annotation techniques be used to improve parsing accuracy? Is there another type of technique not discussed in this paper that could prove useful to unlexicalized PCFGs, perhaps something that has worked for lexicalized PCFGs?