NLP Problem Set 2 – Programming Section
Juliana Louback - jl4354@columbia.edu

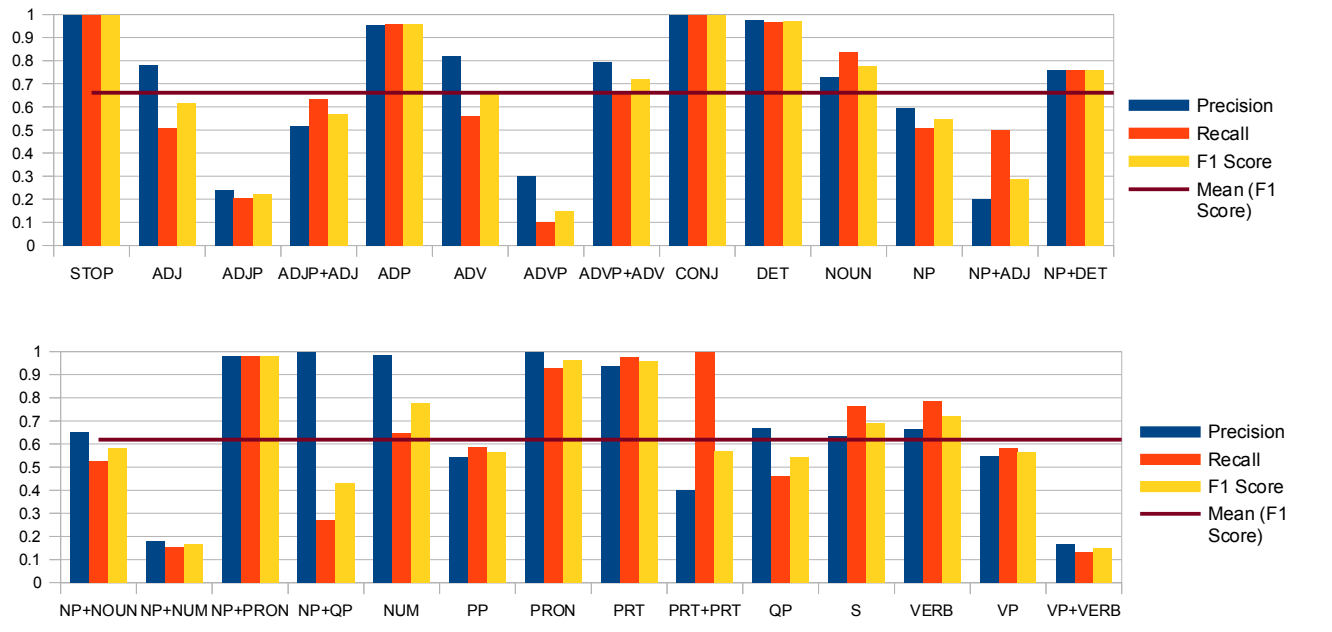
Question 5.

Table 1 displays the performance results of my implementation of the CKY algorithm according to the eval_parser.py program metrics. The model performed quite poorly in both precision and recall on the non-terminals ADJP, ADVP, NP+ADJ, NP+NUM, and VP+VERB; this may be explained by the low frequency of these symbols in the development data, ranging from 2 to 29 when the average frequency is ~173. This is not a hard and fast rule, some words mapped to low frequency symbols such as NP+DET and PRON were still predicted accurately. The precision and indices for the VP and NP nonterminals were suprisingly mediocre, barely above 50%. I can't comment as to why. Figure 1 shows a graph comparing the performance of each nonterminal. The average precision, recall and F1 score of the model over all nonterminals is 70%.

Table 1. Results of CYK algorithm implementation

	Total	Precision	Recall	F1
STOP	370	1	1	1
ADJ	164	0.783	0.506	0.615
ADJP	29	0.24	0.207	0.222
ADJP+ADJ	22	0.519	0.636	0.571
ADP	204	0.956	0.961	0.958
ADV	64	0.818	0.562	0.667
ADVP	30	0.3	0.1	0.15
ADVP+ADV	53	0.795	0.66	0.722
CONJ	53	1	1	1
DET	167	0.976	0.97	0.973
NOUN	671	0.729	0.836	0.779
NP	884	0.594	0.506	0.546
NP+ADJ	2	0.2	0.5	0.286
NP+DET	21	0.762	0.762	0.762
NP+NOUN	131	0.651	0.527	0.582
NP+NUM	13	0.182	0.154	0.167
NP+PRON	50	0.98	0.98	0.98
NP+QP	11	1	0.273	0.429
NUM	93	0.984	0.645	0.779
PP	208	0.542	0.587	0.564
PRON	14	1	0.929	0.963
PRT	45	0.936	0.978	0.957
PRT+PRT	2	0.4	1	0.571
QP	26	0.667	0.462	0.545
S	587	0.634	0.765	0.693
VERB	283	0.665	0.784	0.72
VP	399	0.547	0.581	0.564
VP+VERB	15	0.167	0.133	0.148
total	4664	0.7	0.7	0.7

Figure 1. CYK model performance



Question 6.

I did not find it necessary to change the algorithm used in question 5 for reasons of efficiency. The CKY algorithm implementation for Question 5 ran in about a minute (63 seconds); with vertical markovization, it ran in a little over 2 minues (146 seconds). I used a dynamic lookup table which may explain the efficient runtime. The original algorithm was scalable to a signnificantly higher number of nonterminals. To the right, Table 2 displays the performance results according to the eval_parser.py metrics. Figure 2 (below) contains comparisons of the precision and recall indices of the first and second model, indicated by Precision 1, Precision 2 and Recall 1, Recall 2. Node that the second model has an additional nonterminal SBAR; it not used in the first model and as such is not included in the comparison.

In almost all cases, the second model using vertical markovization matches or slightly improves in precision. However, there is a significant improvement in both the precison and recall of the VP+VERB category. Based on anecdotal knowledge, this improvement is coherent with the application of vertical markovization as it gives more importance to context which is likely to influence a verb. In three cases, ADV, PRT+PRT and PRT the precision decreased slightly using the second model. However, only in the case of ADV and PRT were the recall levels inferior using the second model. The second model has 72% precision, 72% recall and 72% F1 score.

Table 2. Results of CKY algorithm with vertical markovization

	Total	Precision	Recall	F1Score
STOP	370	1	1	1
ADJ	164	0.829	0.622	0.711
ADJP	29	0.281	0.31	0.295
ADJP+ADJ	22	0.417	0.455	0.435
ADP	204	0.96	0.941	0.95
ADV	64	0.765	0.609	0.678
ADVP	30	0.333	0.167	0.222
ADVP+ADV	53	0.72	0.679	0.699
CONJ	53	1	1	1
DET	167	0.976	0.976	0.976
NOUN	671	0.77	0.854	0.81
NP	884	0.612	0.52	0.562
NP+ADJ	2	0.25	0.5	0.333
NP+DET	21	0.842	0.762	0.8
NP+NOUN	131	0.679	0.565	0.617
NP+NUM	13	0.3	0.231	0.261
NP+PRON	50	0.98	0.98	0.98
NP+QP	11	1	0.364	0.533
NUM	93	1	0.667	0.8
PP	208	0.595	0.601	0.598
PRON	14	1	0.929	0.963
PRT	45	0.915	0.956	0.935
PRT+PRT	2	0.333	1	0.5
QP	26	0.765	0.5	0.605
S	587	0.658	0.782	0.714
SBAR	25	0.429	0.24	0.308
VERB	283	0.693	0.813	0.748
VP	399	0.586	0.614	0.6
VP+VERB	15	0.538	0.467	0.5
total	4664	0.721	0.721	0.721

Figure 2. Precision and recall comparison of models