Sanjana Agarwal

ASSIGNMENT #4

1 a) Q : {0,1,2,3} (Set of states)
Σ = {a,b,c} (A finite set of symbols)
q_0 = {0} (The starting state)
F = {0,1,2} (The final states)
δ = Transition function or the transition matrix between states

STATE	**INPUT**		
	a	b	c
0	3	1	2
1	3	1	-
2	2	-	-
3	3	1	-

b) The regex we can use is: ^$(ca*)*$(b+aa*b)+$(b+$)(aa*b)+$((aa*b)+(ab)*)$

2) According to the rule of Markov Chains, number of busy lines will depend only on the number of lines that were busy the last time we observed them, and not on the previous history.
So,

$P_{ij}^{(4)}$ = $P(\xi_{t+4} = S_i \mid \xi_t = S_j)$

= $P(\xi_{t+4} = S_i \mid \xi_{t+3} = S_k) * P(\xi_{t+3} = S_k \mid \xi_{t+2} = S_m) * P(\xi_{t+2} = S_m \mid \xi_{t+1} = S_n) * P(\xi_{t+1} = S_n \mid \xi_t = S_j)$

$= \sum_{r=1}^{n} p_{ik}\, p_{km}\, p_{mn}\, p_{nj}$

Thus, we need to find,
P^4 = P*P*P*P
P^4 =

0.1894	0.3363	0.2721	0.2022
0.1893	0.3364	0.2721	0.2022
0.1821	0.3243	0.2782	0.2154
0.1725	0.3075	0.2853	0.2347

We have been given the initial vector which is,
v = [0.5 0.3 0.2 0.0].
Thus, the answer is, v P^4 = [0.18791 **0.33393** 0.27332 **0.20484**]

(a) Thus, the probability that after 4 steps exactly 3 lines are busy, is, 0.20484
(b) **1 line** will be busy that also has the highest probability(0.33393) after 4 steps.

3) Query: time flies like an arrow
Tagging: time/JJ flies/NNS like/IN an/DT arrow/NN

Thus for bigram 'time flies', we get,
P(NNS|JJ) = 2.4383
For bigram 'flies like', we get,
P(IN|NNS) = 21.8302
For bigram 'like an', we get,
P(DT|IN) = 31.4263
For bigram 'an arrow', we get,
P(NN|DT) = 38.0170
Thus, we get,
Start -> time -> flies -> like -> an -> arrow -> End
Start -> JJ -> NNS -> IN -> DT -> NN -> End

4) The paper presents a statistical model which has the training data of a corpus annotated with POS tags and assigns them to previously unseen text. They use the "Maximum Entropy" model. The probability model they use is defined over H X T, where H is the set of possible word and tag contexts, or "histories", and T is the set of allowable tags. The MaxEnt formalism while testing, uses beam size of N = 5, where increasing the value of N further, does not result in any changes. The model's probability of a history h together with a tag t is defined as: p(h,t) = $\Pi\mu$ $\prod_{j=1}^{k} \alpha_j^{f_j(h,t)}$. The parameters are then chosen to maximize the likelihood of the training data and this formula is extended to the Maximum Entropy formalism. They calculated the model's feature expectation and the observed feature expectation but since the value of H is very large, they use the following expectation. $E(f_j) \approx \sum_{i=1}^{n} \hat{p}(h_i)\, p(t_i|h_i)\, f_j(h_i, t_i)$, where $\hat{p}(h_i)$ is the observed probability of the history h_i in the training set. It scans through (h_i, t_i) and gives some value if the value is not rare (A word that occurs less than 5 times in the training set) $w_i = X$ or it puts x = prefix and suffix where $|X| \leq 4$. It also checks for hyphens and no uppercase and then assigns t_{i-1} = X, $t_{i-2}t_{i-1}$ = XY, $w_{i-1} = X$, $w_{i-2} = X$, $w_{i+1} = X$ and $w_{i+2} = X$ (all places, $t_i = T$). After comparison with other algorithms, the convergence of the accuracy rate implied that any corpus based algorithm on the Penn Treebank Wall St. Journal corpus will not have an accuracy higher than 96.5% due to consistency problems. Thus, the technique they used (Maximum entropy model) is an extremely flexible technique for linguistic modelling. They achieved an accuracy of 96.6% on the unseen test set. This same model however did not perform better than the baseline model on specialized features but much better when used on single annotator features.