Assignment 4 Solution and Marking Scheme

1. Part-of-speech tagging (64 pts)

(a) 8 pts (2 pts for model a), 3 pts for model b), 3 pts for model c))
Model a): conditional distributions: P($T_{i+1} \mid T_i$), P($W_i \mid T_i$), p(T_1)
Joint distribution: Pr(T, W) = p(T_1) $\prod_i P(T_{i+1} \mid T_i) P(W_i \mid T_i)$

Model b): potentials: f(T_i, W_i), f(T_{i+1}, T_i)
Joint distribution: Pr(T, W) = $\frac{1}{k} \prod_i f(T_i, W_i) f(T_{i+1}, T_i)$

Model c): potentials: f(T_i, W_i), f(T_{i+1}, T_i), f(W_{i+1}, W_i)
Joint distribution: Pr(T, W) = $\frac{1}{k} \prod_i f(T_i, W_i) f(T_{i+1}, T_i) f(W_{i+1}, W_i)$

(b) 8 pts (2 pts for model a), 3 pts for model b), 3 pts for model c))
Model a) T*(T-1) + T*(W-1) + (T-1)
Model b) T*T + T*W
Model c) T*T + W*W + T*W

(c) 8 pts (-2 pts per incorrect answer)
Model b) subsumes model a); Model c) subsumes model b) and model a).
If the models are not stationary, model a) also subsumes model b).

(d) 8 pts (2 pts for model a), 3 pts for model b), 3 pts for model c))
Model a): This model is the least expensive in computation for learning the parameters. However, it assumes a causal relationship between tag and word, which may not always be the case. Correlation relations are more appropriate for part-of-speech tagging.

Model b): Computation is not as expensive as model c) for learning the parameters. It also specifies correlation relation between tag and word. However, it does not specify correlations between words.

Model c): Computation is the most expensive. The advantage is that it also specifies correlations between words.

(e) 8 pts (4 pts for each model)
Same for both models
Potentials: f(T_{i+1}, T_i),f(T_i, W_i),
Conditional distribution: P(T | W) = $\frac{1}{k(w)} \prod_i f(T_i, W_i) f(T_{i+1}, T_i)$

(f) 8 pts

The advantage of conditional random fields over Markov networks is that the conditional random fields does not need to model distribution over inputs/words

(g) 8pts (2 pts for predicates, 3 for model b) and 3 for model c))
Tag t = {noun, verb, adjective, adverb, …}
Word w = {the, mountain, is, high, …}
Position p = {1, 2, 3, …}

Predicates:
IsWord(w!, p)
IsTag(t!, p)

First-order formula for model b)
IsTag(+t, p) ^ IsTag(+t', p+1)
IsTag(+t, p) ^ IsWord(+w, p)

First-order formula for model c)
IsTag(+t, p) ^ IsTag(+t', p+1)
IsTag(+t, p) ^ IsWord(+w, p)
IsWord(+w, p) ^ IsWord(+w', p+1)

(h) 8 pts
Markov logic networks provides a compact representation for its corresponding Markov networks.

2. Collective text categorization (36 pts)

(a) 8 pts (2 pts for each answer)
HasWord (word, page): W*P
Topic (class, page): C*P
LinkTo (linked, page, page): L*P*P

It is worthwhile to use a Markov logic network

(b) 8 pts
W*C + 1

(c) 8 pts (4 pts for each rule)
Rule to encode that two pages that have link pointing to the same page are likely to have the same class
Topic(+c,p1)^LinkTo(id1,p1,p2)^LinkTo(id2,p3,p2) => Topic(+c,p3)

Rule to encode that two pages pointed to by links from the same page are likely to have the same topic
Topic(+c,p2)^LinkTo(id1,p1,p2)^LinkTo(id2,p1,p3) => Topic(+c, p3)

(d) 12 pts (4 points for declaration and predicates, 4 pts for each rule)
predicates: isAnchor(id,word, page); isNeighbour(id,word, page)

Some example rules:
The anchor text effects the classification of the topic
Topic(+c,p1)^isAnchor(id1,+w,p1)^LinkTo(id1,p1,p2) => Topic(+c,p2)

The word in the neighbour text affects the weight of relating the topics of pages
Topic(+c,p1)^isNeighbour(id1,+w,p1)^LinkTo(id1,p1,p2) => Topic(+c,p2)