1. (True or False). Wordnet contains relations that constitute a class inheritance hierarchy over the words.

True

2. Give examples of three different types of structural ambiguities and say why they are ambiguous.

PP attachment
John ate the fish from a paper box. Did he eat from a paper box? Or, did he choose the fish from the paper box?

Coordination
John likes red cars and toys. Are the toys red?

NP bracketing
Spanish language teachers: Do the teachers teach Spanish or do they speak Spanish?

\

3. In each of the following sentences, identify the semantic roles selecting from *agent, patient, theme, experiencer, stimulus, goal, recipient, benefactive, source, instrument, location, temporal*. Justify your choice.

The company wrote me a letter.
Jack opened the lock with a paper clip.

The company wrote me a letter.
Agent: the company, recipient: me, patient: letter
Jack opened the lock with a paper clip.
Agent: Jack, patient: lock, instrument: paper clip

4. Consider the following (inelegant) grammar rules from the Penn Treebank:

VP → VBD PP
VP → VBD PP PP
VP → VBD PP PP PP
VP → VBD PP PP PP PP
...

a) **(5 points)** Give two rules that can be used to replace these four (and all the ensuing ... rules).

VP → VBD PP
VP → VP PP

5. (5 points) **True/False:** Given a sentence *S*, the following equation adequately describes the computation performed by the probabilistic CKY algorithm given in the text.

$$P(S) = \sum_{T s.t. S=Yield(T)} P(T)$$

True, the probability of a sentence is equal to the sum of probabilities of all possible parse trees of this sentence. A possible parse tree is any tree that yields the sentence (sequence of tokens).

6. Assuming the grammar below, show the parse tree for the sentence *The big yellow dog sat under the house.*

S-> NP VP
VP -> VP PP
VP -> verb NP
VP -> verb
NP -> DET NOM
NOM -> ADJ NOM
NOM -> NOUN
PP -> PREP NP
DET -> the
ADJ -> big
ADJ -> yellow
NOUN -> dog
VERB -> sat
PREP -> under
NOUN -> house

(S ((NP (DET the) (NOM ((ADJ big) (NOM ((ADJ yellow) (NOM (NOUN dog)))))))
(VP ((VERB sat) (PP ((PREP under)(NP ((DET the) (NOM (NOUN house)))))))

7. Show how you would have to modify the grammar above to handle the sentence *The dog in the white hat ran under the house.*

NOM -> NOM PP

8. **Hidden Markov Models:** You are given the sentence below and the tables of probabilities show in Table 3a (this page) and Table 3b (next page).

I promise to back the bill.

a. Describe how the Table 3a probabilities would be obtained using the Penn Treebank.

count(word,POS)/count(POS). E.g., count(Promise as a VB)/Count(VB)

b. A hidden markov model includes states, observations, transition probabilities, observation likelihoods. Describe what each one of these would correspond to when using an HMM for POS tagging.

states = POS tags
observations = words
transition probabilities = probability of next tag given previous tag
Observation likelihoods = probably of word given a tag.

c. Given the sentence ``I *promise to back the bill.*'' show how you would compute the probability of ``*back*'' as a verb versus the probability of ``*back*'' as a noun using the probabilities in Tables 3a and 3b using the Viterbi algorithm. You are given the values for the third column of the Viterbi table which correspond to observation 3 or ``to''. They are VB: 0, TO: .00000018, NN: 0, PPSS: 0. Thus, you will show two computations both of which will use these values. You do not need to do the arithmetic; just show the formula that would be computed.

Answer: Take the max of four different computations for each current state. We are interested here in the values for two current states: VB and NN. For each of these, three computations will be 0 since the observations from the three (VB, NN, PPSS) previous states are 0.

For VB, the computation for the remaining path from TO is: $vt_{i-1}(i)*a_{ij}*b_j(o_t)$ = .00000018*.83*.0008

For NN, the computation for the remaining path from TO is: .00000018*.00047*.00068

If we multiply out, we see that probability as VB is greater. For VB, .00000000011952. For NN, .00000000000006.

	I	*promise*	*to*	*back*
VB	0	.0093	0	.00008
TO	0	0	.99	0
NN	0	.0085	0	.00068
PPSS	.37	0	0	0

Table 3a: Observation Likelihoods

	VB	**TO**	**NN**	**PPSS**
<s>	.019	.0043	.041	.067
VB	.0038	.035	.047	.0070
TO	.83	0	.00047	0
NN	.0040	.016	.087	.0045
PPSS	.23	.00079	.0012	.00014

Table 3b: Tag transition probabilities. The rows are labeled with the conditioning event. Thus, P(VB|<s>) = .019.

9. Consider the sentences *President George Bush has re-invigorated the economy by providing a bail-out program for failing Wall Street firms*. and *President George Bush has caused a disastrous economic situation by failing to provide regulations on Wall Street firms.* You'd like to compute the likelihood of these sentences given a corpus of NY Times, Wall Street Journal and the New York Post gathered over the last year. You develop a bi-gram language model. Describe how you would: 1. Build the language model, 2. Compute the likelihood of these sentences and 3. Evaluate your language model.

I would build the language model by computing the probability of each bigram (pair of words) in the corpus. To do this, I would count the frequency of each pair of words w1 and w2 and I would then normalize by the count of the first word in the bigram: Count(w1 w2)/ Count(w1). I would do this for each pair of words that appears consecutively in the corpus. Once I had the language model, I would compute the likelihood of each sentence by determining the probability of each bigram in the sentence (including the probability of start followed by word-1) and multiplying. So, for eaxmaple, Prob(start President) * (President George) * Prob (George Bush) * Prob (Bush has).. etc. This must be done for both sentences. There are several ways to evaluate the language model.