Final Exam
Natural Language Processing

1st of June 2015

Remember to fill in your name on all pages
Good luck

Problem 1 (1 point). Compute the Minimal Edit Distince between "drive" and "brief" using dynamic programming.

Problem 2 (1 point). What is the mean average precision (MAP) for the following sequence of retrieved documents, where R denotes a relevant document and N denotes an irrelevant document? (Assume there are 20 relevant documents in the collection)

R R N R N N R N N N R N R N R

Problem 3 (1.5 points). Evaluate Labeled Precision, Labeled Recall and F1 score for the sentence "Perhaps you could hasten a prediction of where this is all going .", knowing the parser's guess and the gold answer.

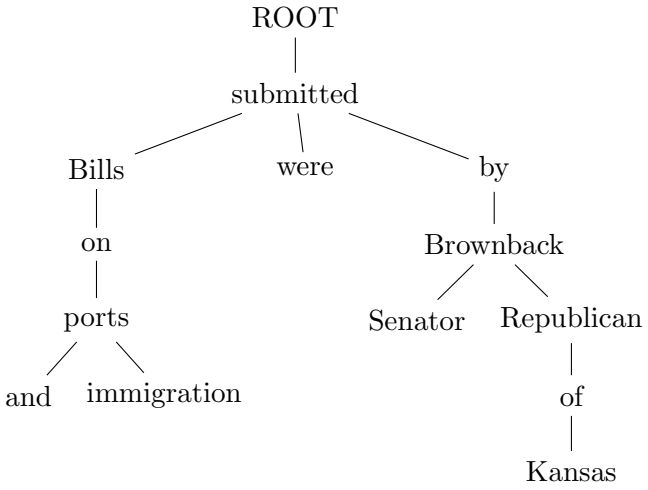
Guess:

```
(ROOT
 (VP (RB Perhaps)
  (VP (PRP you)
   (VP (MD could)
    (VP (LS hasten)
     (VP (DT a)
      (VP (LS prediction)
       (VP (IN of)
        (VP (WRB where)
         (NP (DT this)
          (NP (VBZ is)
           (NP (PDT all)
            (NP (VBG going) (. .)))))))))))))
```

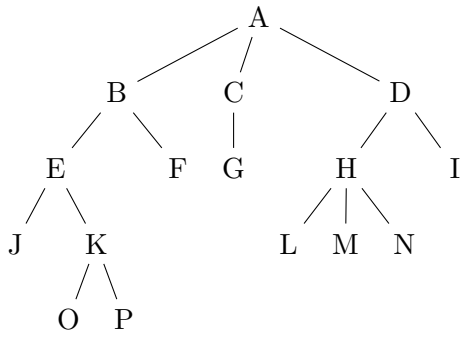
Gold:

```
(ROOT
 (S
  (ADVP (RB Perhaps))
  (NP (PRP you))
  (VP (MD could)
   (VP (VB hasten)
    (NP
     (NP (DT a) (NN prediction))
     (PP (IN of)
      (SBAR
       (WHADVP (WRB where))
       (S
        (NP (DT this))
        (VP (VBZ is) (DT all)
         (VP (VBG going)))))))))
  (. .)))
```

Problem 4 (2.5 points). Given the English sentence "Bills on ports and immigration were submitted by Senator Brownback Republican of Kansas", simulate the run of the "Arc-eager" Dependency Parser in order to get its dependency tree (which you can find below).



Problem 5 (1.5 points). Assume we have a corpus of 1000 words and the following WordNet Hierarchy:



Now assume we collect the following count data for each of the words:
J = 100, O = 100, B = 100, F = 100, G = 100, L = 100, M = 100, N = 100, I = 200
Assuming that all other words do not appear in the corpus, what is $sim_{Lin}(B, H)$? Use natural log in your calculation.

Problem 6 (2.5 points). Compute the cosine similarity of "apple" and "plum" using PPMI (Positive Pointwise Mutual Information) given the following table of context frequencies (use add-1 smoothing in your calculations).

word/context	digital	results	sugar	pinch
apple	0	1	3	2
plum	0	0	4	6
computer	5	3	0	0
information	4	5	1	0

DEFINITION 1. A dynamic programming algorithm that computes Minimal Edit Distance: for two words w and w' of length n and m, respectively, we compute $D(i,j)$ for small i,j, and then larger $D(i,j)$ based on previously computed smaller values, where $D(i,j)$ = min edit distance between the i-length prefix of w and the j-length prefix of w'.

- Initialization
  $D(i,0) = i, \forall i \in 0,\ldots,n$
  $D(0,j) = j, \forall j \in 0,\ldots,m$
- Recurrence Relation:

  For each $i = 1,\ldots,n$

  For each $j = 1,\ldots,m$

  $$D(i,j) = \min(D(i-1,j)+1, D(i,j-1)+1, D(i-1,j-1)+d),$$

  where $d = \begin{cases} 2, & \text{if } w_i \neq w'_j \\ 0, & \text{if } w_i = w'_j \end{cases}$
- Termination
  Output $D(n,m)$

DEFINITION 2. The Mean Average Precision (MAP) is the average of the precision value obtained for the top k documents, each time a relevant document is retrieved.

DEFINITION 3. The following measures help evaluate the quality of a parsing algorithm:

Labeled Precision (P):	TP/(TP+FP)
Labeled Recall (R):	TP/(TP+FN)
F1:	2P*R/(P+R)

TP = true positives (the number of gold answers that were correctly guessed)
FP = false positives (the number of guessed answers that were incorrect)
FN = false negative (the number of gold answers that were not correctly guessed)

DEFINITION 4. The "Arc-eager" Dependency Parser:

Start: σ = [ROOT], $\beta = w_1,\ldots,w_n$, $A = \emptyset$

1. Shift	$\sigma, w_i \mid \beta, A$	$\sigma \mid w_i, \beta, A$
Precondition:	$r'(w_k, w_i) \notin A$,	$w_i \neq$ ROOT
2. Left-Arc$_r$	$\sigma \mid w_i, w_j \mid \beta, A$	$\sigma, w_j \mid \beta, A \cup \{r(w_j, w_i)\}$
3. Right-Arc$_r$	$\sigma \mid w_i, w_j \mid \beta, A$	$\sigma \mid w_i \mid w_j, \beta, A \cup \{r(w_i, w_j)\}$
Precondition:	$r'(w_k, w_i) \in A$	
4. Reduce	$\sigma \mid w_i, \beta, A$	σ, β, A

Finish: $\beta = \emptyset$

DEFINITION 5. Let $words(c)$ be the set of all words that are children of node c (including itself)

$$P(c) = \frac{\sum_{w \in words(c)} count(w)}{N}$$

(N = total number of tokens in the corpus)

$$IC(c) = -\log P(c)$$

$LCS(c_1, c_2)$ = The most informative (lowest) node in the hierarchy subsuming both c_1 and c_2

$$sim_{Lin}(c_1, c_2) = \frac{IC(LCS(c_1, c_2))}{IC(c_1) + IC(c_2)}$$

Definition 6. Computing PPMI on a Term-context Matrix

Matrix F with W rows (words) and C columns (contexts)

f_{ij} is the number of times w_i occurs in context c_j

$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$

$p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$

$p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$

$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*}p_{*j}}$

$ppmi_{ij} = \max(pmi_{ij}, 0)$

Add-one smoothing is pretending to have seen each pair (word, context) one more time.

$cos(\vec{v}, \vec{w}) = \frac{\vec{v} \bullet \vec{w}}{|\vec{v}||\vec{w}|} = \frac{\vec{v}}{|\vec{v}|} \bullet \frac{\vec{w}}{|\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$

v_i is the PPMI value for word v in context i

w_i is the PPMI value for word w in context i.

$cos(\vec{v}, \vec{w})$ is the cosine similarity of v and w