Speech and Natural Language Processing
Autumn Semester, 2016
Maximum Marks: 40

CS60057
Mid-Sem
Time Limit: 2 Hours

This exam contains 2 pages (including this cover page) and 7 problems. There are no clarifications. In case of doubt, you can take a valid assumption, state that properly and continue.

1. (5 points) Zipf's law provides a relation between the frequency of a word and its rank. However, multiple words may have the same frequency, which are given different ranks. Let n denote the number of times a word appears in the collection. Let us define $MaxRank(n)$ as the maximum rank obtained by any of the words with frequency n. For example, if $n = 90$, and the words, 'in', 'and', 'said' apprear in the corpus with the frequency $n = 90$, and their ranks are 5, 6 and 7, respectively, then $MaxRank(90) = 7$.

   Now, suppose that Zipf's law holds for frequency n and $MaxRank(n)$. Can you estimate the number of words that occur exactly n times? You are given that V is the size of the vocabulary.

2. (6 points) We discussed the dynamic programming formulation of the edit distance computation. Explain

   (a) (3 points) How would you modify the recurrence relation to incorporate transposition of letters as an additional edit operation? $transpose(x, y) = (y, x)$

   (b) (3 points) Suppose the following table provides the common substitution errors encountered in an English corpus. The element (m, n) of this table denotes – how often is an incorrect letter m substituted for a correct letter n. Provide a suitable formulation of a cost matrix to be used for the weighted edit distance. Assume there are only 5 characters in your alphabet.

	a	b	c	d	e
a	0	0	7	1	23
b	0	0	9	9	2
c	6	5	0	16	0
d	1	10	13	0	12
e	27	0	3	11	0

3. (4 points) Suppose you typed 'hoper' by mistake instead of 'hotel'. Explain how would the symmetric delete spelling correction would work to provide the correct candidate 'hotel' but not words like 'horse', 'chore' etc.

4. (6 points) Consider the maximum entropy model for Named Entity Recognition, where you want to estimate $P(tag|w_i)$. In a hypothetical setting, assume that *tag* can take the values *L*, *D*, *P* and *N* (short forms for location, drug, person and none). The variable *word* w_i could be any member of a set V of possible words. The distribution should give the following probabilities

   - $P(L|w_{i-1} = \text{'}in\text{'} \text{ and } isCapitalized(w)) = 0.9$
   - $P(D|ends(w, \text{'}c\text{'})) = 0.9$
   - $P(P|ends(w_{i+1}, \text{'}ed\text{'})) = 0.7$

- $P(N|w_i = \text{`}in\text{'}) = 0.8$
- $P(N|ends(w_i, \text{`}ed\text{'})) = 0.6$

It is assumed that all other probabilities, not defined above could take any values such that $\sum_{tag} P(tag|w_i) = 1$ is satisfied for any word in V.

(a) (3 marks) Define the features of your maximum entropy model that can model this distribution. Mark your features as f_1, f_2 and so on. Each feature should have the same format as explained in the class. [**Hint:** 5 Features should make the analysis easier]

(b) (3 marks) For each feature f_i, assume a weight λ_i. Consider the following sentence: 'Sue purchased Zantec in Arcadia'. For this sentence, write expression for the following probabilities in terms of your model parameters

- $P(P|Sue)$
- $P(N|purchased)$
- $P(D|Arcadia)$

5. (6 points) In a corpus, suppose there are 5 words, a, b, c, d and e. You are provided with the following counts.

n-gram	count	n-gram	count	n-gram	count
bca	0	ca	2	a	8
bcb	2	cb	4	b	10
bcc	0	cc	0	c	6
bcd	3	cd	4	d	12
bce	0	ce	0	e	4

Use the recursive definition of backoff smoothing provided in the class to obtain the probability distribution, $P_{backoff}(w_n|w_{n-1}w_{n-2})$, where $w_{n-1} = c$ and $w_{n-2} = b$.

Also assume that $\hat{P}(x) = P(x) - 0.1$.

6. (5 points) For a news corpus, the number of unique words were found to be $200,000$. The number of unique bigrams were found to be $4,000,000$, out of which 50% occurred only once, while 20% occurred twice. Suppose you are using Good-Turing smoothing. Estimate the effective bigram count for $c = 0$ and $c = 1$ using Good-Turing smoothing, where c denotes the count observed in the news corpus.

7. (8 points) Consider the following simple PCFG:

S → NP VP	0.6	PropNoun → DALLAS	0.2
S → VP	0.4	PropNoun → ALICE	0.2
NP → NP PP	0.4	PropNoun → BOB	0.3
NP → PropNoun	0.6	PropNoun → AUSTIN	0.3
VP → Verb	0.3	Verb → ADORE	0.5
VP → Verb NP	0.3	Verb → SEE	0.5
VP → VP PP	0.4	Prep → IN	0.4
PP → Prep NP	1.0	Prep → WITH	0.6

(a) (2 marks) Convert the above PCFG into Chomsky Normal Form.

(b) (6 marks) Use the inside algorithm to find the probability for generating the sentence: *SEE BOB IN AUSTIN*