CS6320, Fall 2017
Dr. Mithun Balakrishna
Homework 2
Due October 1st, 2017 11:59pm

A. Submission Instructions:

- Submit your solutions via eLearning.
- Please submit a single zip file with the following files:
  - For programming questions:
    - Source code file(s) in C/C++, Java, or Python. For using any other programming language, please get prior approval from the TA.
    - A ReadMe file with instructions on how to compile/run the code.
  - For all other questions, a PDF/Doc/PS/Image file with the solutions.
- Late Submission Penalty:
  - up to 2 hours late — 10% deduction
  - 2 - 4 hours late — 20% deduction
  - 4 - 12 hours late — 35% deduction
  - 12 - 24 hours late — 50% deduction
  - 24 - 48 hours late — 75% deduction
  - more than 48 hours late — 100% deduction (zero credit)

B. Problems:

1. Bigram Probabilities (45 points):

An automatic speech recognition system has provided a written sentence as the possible interpretation to a speech input.

Compute the probability of a written sentence using the bigram language model trained on *HW2_F17_NLP6320-NLPCorpusTreebank2Parts-CorpusA.txt* (provided as Addendum to this homework on eLearning).

Note: Please use whitespace to tokenize the corpus into words that are required for the bigram model.

Compute the sentence probability under the three following scenarios:

i. Use the bigram model without smoothing.
ii. Use the bigram model with add-one smoothing
iii. Use the bigram model with Good-Turing discounting.

Your computer program should do the following:

1. Compute the bigram counts on the given corpus (*HW2_F17_NLP6320-NLPCorpusTreebank2Parts-CorpusA.txt*).
2. For a given input written sentence:

a. For each of the three scenarios, construct a table with the bigram counts for the sentence.

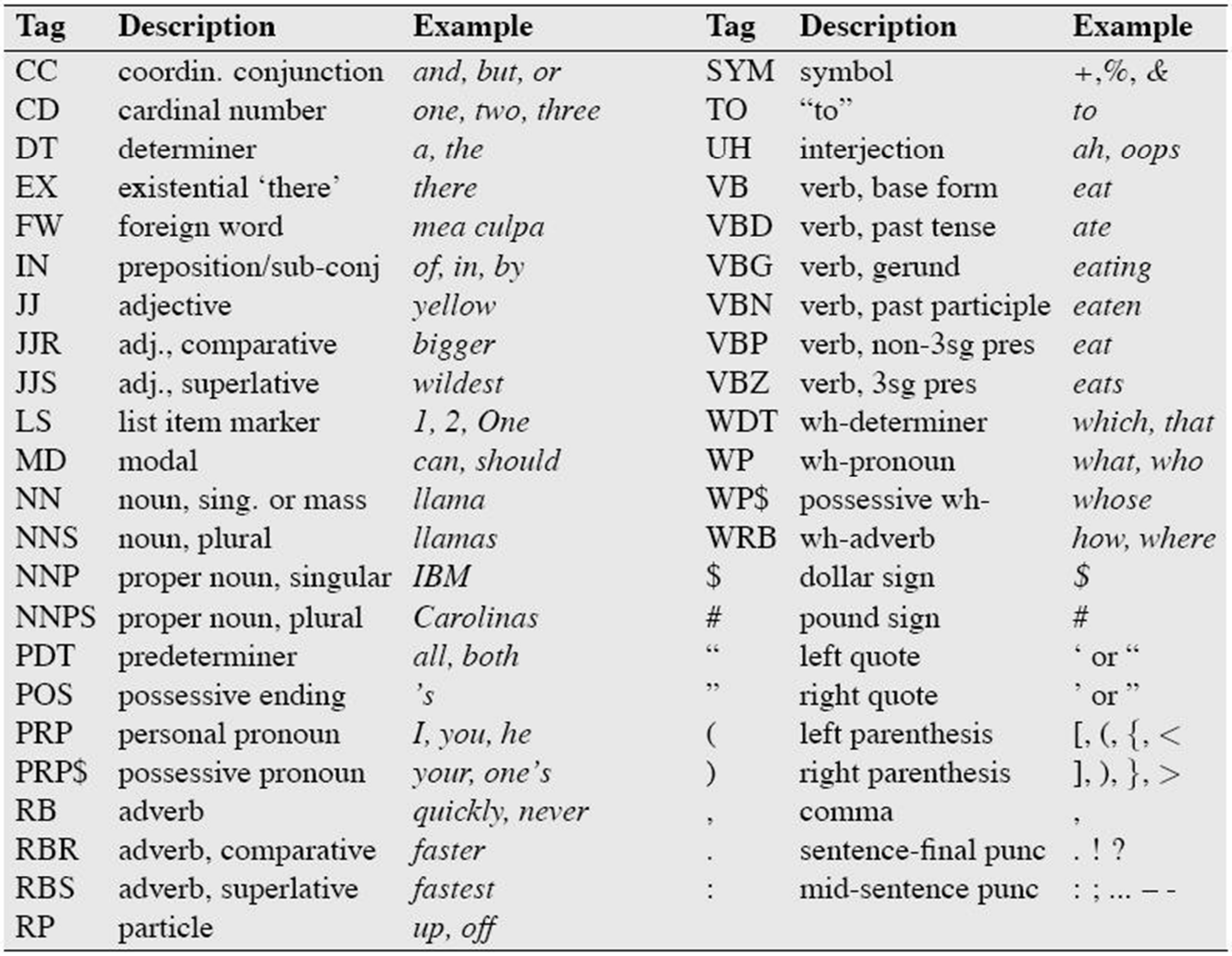
b. For each of the three scenarios, construct a table with the bigram probabilities for the sentence.

c. For each of the three scenarios, compute the total probability for the sentence.

2. POS Tagging Errors (10 points)

Find one tagging error in each of the following sentences that are tagged with the Penn Treebank POS tagset (Figure 1):

1. I/PRP need/VBP a/DT flight/NN from/IN Atlanta/NN
2. Does/VBZ this/DT flight/NN serve/VB dinner/NNS
3. I/PRP have/VB a/DT friend/NN living/VBG in/IN Denver/NNP
4. Can/VBP you/PRP list/VB the/DT nonstop/JJ afternoon/NN flights/NNS



Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two, three*	TO	“to”	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential ‘there’	*there*	VB	verb, base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb, past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb, non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb, 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, singular	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	#
PDT	predeterminer	*all, both*	“	left quote	‘ or “
POS	possessive ending	*’s*	”	right quote	’ or ”
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one’s*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

Figure 1. Penn Treebank POS tagset

3. Transformation Based POS Tagging (45 points)

For this question, you have been given a POS-tagged training file, *HW2_F17_NLP6320_POSTaggedTrainingSet.txt* (provided as Addendum to this homework on eLearning), that has been tagged with POS tags from the Penn Treebank POS tagset (Figure 1). Use this POS tagged file to:

a. Create a unigram model containing the most probable POS tag for each word in the corpus's vocabulary.

   Note: compute this probability by considering each word in isolation. Do NOT use any context (i.e. previous words or tags) for compute the most probable POS tag for a word.

b. Brill's transformation rules: Implement Brill's transformation-based POS tagging algorithm using ONLY the previous word's tag to create transformation rules.

c. Apply model (a) and (b) on the sentence below, and show the difference in error rates.

   The president wants control of the board 's control