

**CS6320, Fall 2017**  
**Dr. Mithun Balakrishna**  
**Homework 2**  
**Due October 1<sup>st</sup>, 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 (i.e. space, tab, and newline) to tokenize the corpus into words/tokens that are required for the bigram model. Do NOT perform any type of word/token normalization (i.e. stem, lemmatize, lowercase, etc.). Creation and matching of bigrams should be exact and case-sensitive. Do NOT split the corpus into sentences. Please consider the entire corpus as a single string for tokenization and computation of bigrams.**

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

**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.

**Sentence:** *The president wants to control the board 's control*

**POS Tagged Sentence:** *The\_DT president\_NN wants\_VBZ to\_TO  
control\_VB the\_DT board\_NN 's\_POS control\_NN*