CS6320, Spring 2020
Dr. Mithun Balakrishna
Homework 3
Due Sunday, March 8th, 2020 11:59pm

A. Submission Instructions:

- Submit your solutions via eLearning.
- Please submit a single zip file containing **ALL** the relevant homework solution files. The zip filename should follow the pattern "HW#_FirstnameLastname.zip" (Example: HW3_Claire Underwood.zip)
    - **Penalty of 5 points** if not followed
- For all non-programming questions:
    - Please include **ALL** the solutions in a **single** PDF/Doc/PS/Image file
    - The filename should follow the pattern "HW#_FirstnameLastname.FileExtension" (Example: HW3_Claire Underwood.pdf)
    - **Penalty of 5 points** if not followed
- For programming questions:
    - Write the programming solutions in C/C++, Java, or Python. For using any other programming language, please get prior approval from the TA.
    - Include a Readme file with instructions on how to build and run your programming question solution
        - Instructions should be very simple:
            - python bigram.py input_arguments

              OR

              python bigram.py (if the input arguments are hard coded)
        - Hard coding the input arguments to your program is fine unless the TA cannot run your code directly. Do **NOT** include instructions such as: "Please modify the path in my main function. Then copy the training data in the same folder."
        - Provide your training data together unless the dataset is too large.
        - **Penalty of 10 points** if not followed
    - Submit ALL your source code files
        - Do not write your solutions in the readme file
        - **Penalty of 10 points** if not followed
- 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. Probabilistic POS Tagging (45 points)

For this question, you have been given a POS-tagged training file, *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).

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

Use the POS tagged file to perform **Naïve Bayesian Classification (Bigram) based POS Tagging**:

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n) \approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1})$$

- Using the given corpus, write a computer program to compute the bigram models (counts and probabilities) required by the above Naïve Bayesian Classification

formula. Please do not submit the bigram models with the homework solution submission. The TA can run your program to produce and check them.

- Using the created bigram models, write a computer program to compute the POS tags for any input sentence using the above Naïve Bayesian Classification formula. Your program should assign a POS tag for every space separated word in an input sentence.

Other Instructions:

1. Use each line (ending with newline character) in the corpus as a single text sentence.

2. Use whitespace (i.e. space, tab, and newline) to tokenize each text sentence into words/tokens.

3. Use the WORD_POS pattern to extract the actual word and part-of-speech tag (i.e. the WORD part in the WORD_POS pattern is the actual word and POS part in the WORD_POS pattern is the part-of-speech tag) from the tokenized word.

   For example, in the tokenized word "Brainpower_NNP", "Brainpower" is the actual word and "NNP" is the part-of-speech tag.

4. Statistical model N-Grams should be considered ONLY within a text sentence.

5. Do NOT perform any type of word/token normalization (i.e. case-normalization, stemming, lemmatization, etc.).

6. Creation and matching of words and part-of-speech tags should be exact and case-sensitive.

7. Bigram smoothing is NOT required.

8. Please consider special tag "<s>" at the start of a text sentence and "</s>" at the end of a text sentence. The formula to compute the POS tag sequence for the input sentence "*John went to work .*" will be:

$$\hat{t}_1^5 = \underset{\hat{t}_1^5}{\operatorname{argmax}} P(John|tag_1) * P(tag_1| < s >) * P(went|tag_2) * P(tag_2|tag_1) * P(to|tag_3) * P(tag_3|tag_2) * P(work|tag_4) * P(tag_4|tag_3) * P(.|tag_5) * P(tag_5|tag_4) * P(</s > |tag_5)$$

2. HMM Decoding: Viterbi Algorithm (45 points):

Π →	NNP	MD	VB	JJ	NN	RB	DT
<s>	0.2767	0.0006	0.0031	0.0453	0.0449	0.0510	0.2026
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	0.0090	0.0025
MD	0.0008	0.0002	0.7968	0.0005	0.0008	0.1698	0.0041
VB	0.0322	0.0005	0.0050	0.0837	0.0615	0.0514	0.2231
JJ	0.0366	0.0004	0.0001	0.0733	0.4509	0.0036	0.0036
NN	0.0096	0.0176	0.0014	0.0086	0.1216	0.0177	0.0068
RB	0.0068	0.0102	0.1011	0.1012	0.0120	0.0728	0.0479
DT	0.1147	0.0021	0.0002	0.2157	0.4744	0.0102	0.0017

Table 1. HMM Transition Probability

	Janet	will	back	the	bill
NNP	0.000032	0	0	0.000048	0
MD	0	0.308431	0	0	0
VB	0	0.000028	0.000672	0	0.000028
JJ	0	0	0.000340	0	0
NN	0	0.000200	0.000223	0	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0

Table 2. HMM Observation Likelihood

For the HMM shown above, please perform the following:

Programmatically implement the Viterbi algorithm to compute the most likely tag sequence and its probability for any given observation sequence. Example observation sequences:

i. *Janet will back the bill*
ii. *will Janet back the bill*
iii. *back the bill Janet will*

3. Maximum Entropy Modeling (10 points):

Consider the following Maximum Entropy features and weights:

$$f_1(c,x) = \begin{cases} 1 \text{ if } word_i = \text{"race"} \ \& \ c = \text{NN} \\ 0 \text{ otherwise} \end{cases}$$

$$f_2(c,x) = \begin{cases} 1 \text{ if } t_{i-1} = \text{TO} \ \& \ c = \text{VB} \\ 0 \text{ otherwise} \end{cases}$$

$$f_3(c,x) = \begin{cases} 1 \text{ if } t_{i-1} = \text{DT} \ \& \ c = \text{NN} \\ 0 \text{ otherwise} \end{cases}$$

$$f_4(c,x) = \begin{cases} 1 \text{ if is\_lower\_case}(word_i) = \text{"race"} \ \& \ c = \text{VB} \\ 0 \text{ otherwise} \end{cases}$$

$$f_5(c,x) = \begin{cases} 1 \text{ if } word_i = \text{"race"} \ \& \ c = \text{VB} \\ 0 \text{ otherwise} \end{cases}$$

$$f_6(c,x) = \begin{cases} 1 \text{ if } t_{i-1} = \text{TO} \ \& \ c = \text{NN} \\ 0 \text{ otherwise} \end{cases}$$

		Weights					
		f1	f2	f3	f4	f5	f6
Tags	VB	0	0.75	0	0.10	0.15	0
	NN	0.3	0	0.9	0	0	-0.2

Manually compute the best tag for the word "race" in the following sentences:

a. Secretariat/NNP is/VBZ expected/VBN to/TO **race**/?? tomorrow/NN

b. the/DT **race**/?? for/IN outer/JJ space/NN