Natural Language Processing
Computer Engineering Dept.
Prof. Sandra Wahid

Fourth year
Jan 21st, 2024

Final Exam

Q1 Multiple Choice Questions (just 1 correct choice) [1 point each]:

1. The following statement is **NOT** an advantage of dense vectors to sparse vectors:

a) Easier to use as a feature in other Machine Learning models

b) Handles unknown words unlike long sparse vectors

c) More general representation of rare words

d) Dense vectors are shorter than sparse vectors

2. Encoding bigrams using a one-hot encoding gives a vector representation of size ..

a) |V|

b) $|V^2|$

c) |2V|

d) $|V|^2$

3. (0|1)*000 is a regular expression that matches .....
a) All even binary strings
b) All binary strings that are divisible by 4
c) All binary strings that are divisible by 8
d) All binary strings ending with 00

4. Attention mechanism allows the encoder to get information from all the hidden states of the decoder.
a) True
b) False

5. The accuracy is the same as **Macro** average precision of binary classification.

a) True

b) False

6. The Markov chain representing a tri-gram language model is computed by:

a) $P(w_1) \prod_{i=2}^{n} p(w_i|w_{i-1}, w_{i-2})$

b) $P(w_1)P(w_2|w_1) \prod_{i=2}^{n-1} p(w_{i+1}|w_i, w_{i-1})$

c) $P(w_1)P(w_2|w_1) \prod_{i=1}^{n} p(w_i|w_{i-1}, w_{i-2})$

d) $P(w_2|w_1) \prod_{i=2}^{n} p(w_i|w_{i-1}, w_{i-2})$

$VWf(x)$

7. Consider the equation y = softmax(V ~~g~~(W f (x))), where f (x) is a vector of features extracted over the input x, V and W are parameter matrices, and g is a linear function. Suppose we remove g from this equation (replace it by identity function). What happens?

a) Nothing changes.

b) The model is different but just as powerful as before.

c) The model is more powerful because nonlinearities can throw away information by "squashing" their inputs.

d) The model becomes equivalent to logistic regression

8. For the given features, which training model is the most appropriate to use

	Number of the word "good" in the document	Number of the word "wonderful" in the document
d1	2	1
d2	4	2
d3	8	4
d4	12	6

a) Naive Bayes

b) Logistic Regression

c) Both of them can be used

d) None of them can be used

9. The more you increase n in the n-gram language model, this increase the long-term word dependencies

a) True

b) False

10. For the given feature template, can we use it for linear chain CRF: <y_{i-2}, y_{i-1},y_i,x_{i+1},x_{i+2}>

a)True

b)False

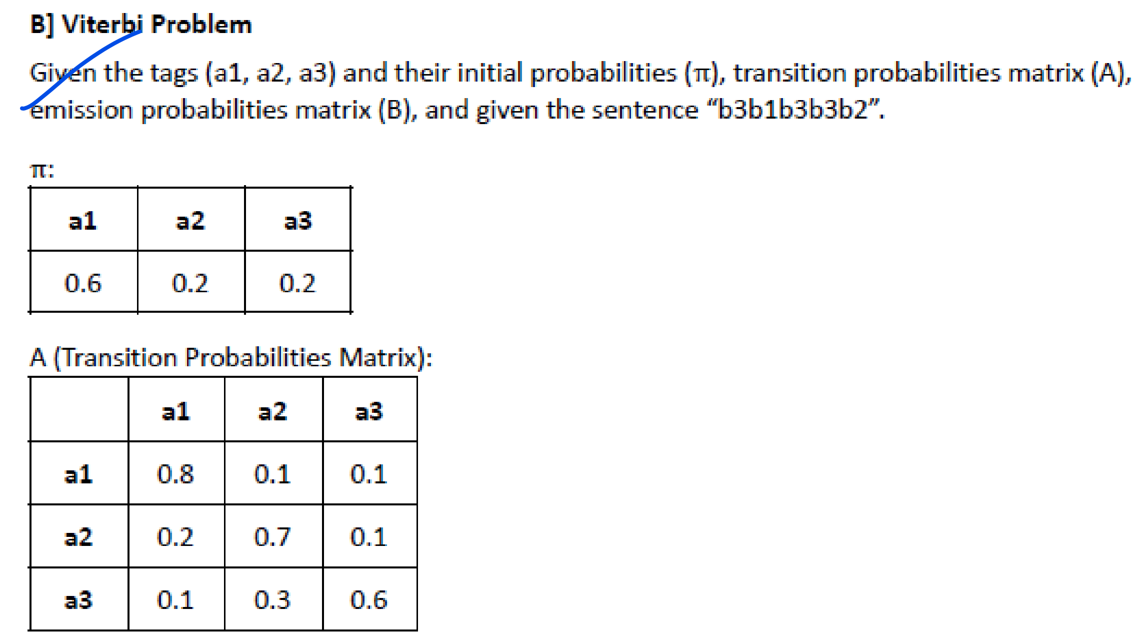
-The following is a weight update equation for the Skip Gram Model

$$w'^{(new)}_{ij} = w'^{(old)}_{ij} - \eta \cdot \cancel{\sum_{c=1}^{C}} (y_{c,j} - t_{c,j}) \cdot h_i$$

Without mathematical proof, derive the same equation for the CBOW model

-You use an RNN to do sentiment analysis on hotel reviews. Each review receives a score of 0 (very negative) to 5 (very positive). The output depends on the last hidden layer only. The RNN was given this input: “The vacation was ruined, the food was bad, the rooms were bad. But surprisingly the staff seemed very happy”. The model outputs “very positive”. Why did this misclassification occur? (Not this input but the question remains the same)

-Viterbi Question as last year (same initial, transmission and emission matrices but the sequence was 6 long instead of 5)



B] Viterbi Problem

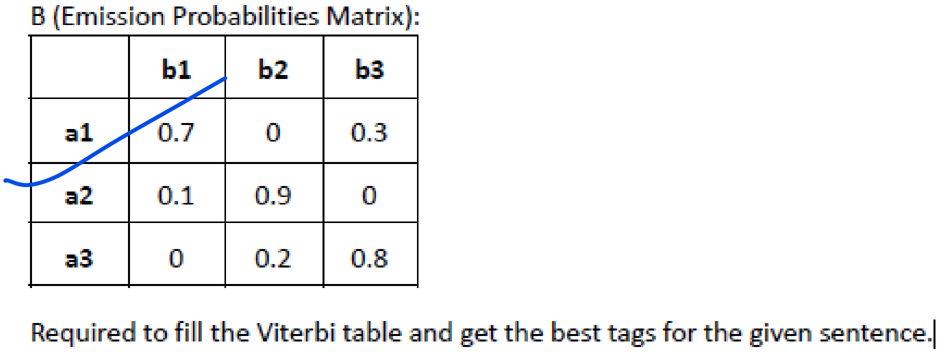
Given the tags (a1, a2, a3) and their initial probabilities (π), transition probabilities matrix (A), emission probabilities matrix (B), and given the sentence "b3b1b3b3b2".

π:

a1	a2	a3
0.6	0.2	0.2

A (Transition Probabilities Matrix):

	a1	a2	a3
a1	0.8	0.1	0.1
a2	0.2	0.7	0.1
a3	0.1	0.3	0.6



B (Emission Probabilities Matrix):

	b1	b2	b3
a1	0.7	0	0.3
a2	0.1	0.9	0
a3	0	0.2	0.8

Required to fill the Viterbi table and get the best tags for the given sentence.

-Having a dataset and you want to train it to get the tf-idf of the document, and you have 3 python functions

1- fit_transform: train the data to get the tf and returns the term-document matrix

2-transform: Calculates and returns the term document matrix

3-fit: train the data to get the tf

Based on the best practices:

a) which function(s) would you use for the training data….. Fit trans

b) which function(s) would you use for the test data….. trans

-The following matrix is the transition matrix between these tags. In the exam you know Nothing about the dataset, yet there are cells that you know their value. fill these cells using **BIO** NER

Note: Don't assume equiprobability the doctor just wanted you to write the zeroes in the matrix

	B_ORG	I_ORG	B_CITY	I_CITY	O
<s>					
B_ORG	for different org it will be != 0				
I_ORG		0		0	
B_CITY					
I_CITY		0		0	
O					

A CBOW word2vec model learns the word vector matrix W_1 , and a co-occurrence count based method also learns another word vector matrix W_2 and a skip-gram model learns the word vector matrix W_3

1. Are W_1, W_2 identical? (Yes/No)
2. Are W_1, W_3 identical? (Yes/No)
3. If the CBOW model was retrained, would the new vector matrix be identical to the old one W_1 ? (Yes/No)
4. If the co-occurrence model was retrained, would the new vector matrix be identical to the old one W_2 ? (Yes/No)

-Regex question

```
text = “This $question$ is a must in the exam”
text = re.sub(r'$', ' :Regex ', text))
text = re.sub(r'\$[a-z]*\$ ', ’’, text))
print(text)
```

-Given a random model and a test set containing a sentence of 20 characters drawn randomly from the English alphabet. (Note that the English alphabet contains 26 characters) Compute the Perplexity of this test sentence (and don’t forget to write the equation).

-Calculate PPMI for “machine” and “car” knowing that they occur 190 and 60 respectively and the total number of words in the document is 800 and “machine” and “car” co-occurred 40 times

-a 2-layer neural network is used for sentiment analysis of hotel reviews, It uses a set of features x to predict one of three sentiments for a review. The distinguishing features between reviews are: The length of the review, the presence of the word “good” in the review, the presence of the word “bad” in the review, the number of words in the review from the positive lexicon, the number of words in the review from negative lexicon.

5

The network has the following equations:

z[1] = W[1]*x + b[1]

a[1] = ReLu(z[1])

z[2] = W[2]*a[2] + b[2]

y = SoftMax(z[2])

1) The previous equations have 1 error, fix the error
2) Deduce the dimensionality of x if possible, if not write (not able to determine) 5feat
3) Deduce the dimensionality of W[2] if possible, if not write (not able to determine) 4
4) Deduce the dimensionality of b[2] if possible, if not write (not able to determine)

3x1

-An encoder-decoder system is used for french machine translation, an example of an input sequence used is “I Love Machine Learning” which was translated into “Je Aime Lapprentissage Automatique”. Keep in mind that **dot-product attention** is used and **teacher forcing** is used, and assume that the function used to calculate the hidden states is a simple **element-wise addition**. The following vectors were given in the question:

1. Input Sequence Embeddings (I don’t remember them, but they were not used):

Embedding[“I”] = [0.1 , 0.2]
Embedding[“Love”] = [0.3 , 0.4]
Embedding[“Machine”] = [0.5 , 0.6]
Embedding[“Learning”] = [0.7 , 0.8]

2. Output Sequence Embeddings:

Embedding[“<s>”] = [0.8 , 0.9]
Embedding[“Je”] = [1.1 , 1.2]
Embedding[“Aime”] = [1.3 , 1.4]
Embedding[“Lapprentissage”] = [1.5 , 1.6]
Embedding[“Automatique”] = [1.7 , 1.8]

3. Hidden States of Encoder (I don’t remember them, but you will need all of them):

He[1] = [0.15 , 0.25]
He[2] = [0.45 , 0.65]
He[3] = [0.85 , 1.05]
He[4] = [1.25 , 1.45]

1) Find the context vector c1
2) Find the decoder hidden state vector Hd[1]
3) Find the context vector c2