CS4248 Midterm

19 February 2021

Please do not turn to the next page until you are told to do so by your proctor.

- This midterm is scored out of a total of **60** marks and is worth 20% of your final marks for the course.
- This midterm is estimated to take you about **60** minutes to complete.
- This midterm has a total of **13** questions.
- You can visit `http://www.comp.nus.edu.sg/~cs4248/2020/midterm.html` to reach the entry form in LumiNUS if you lose your browser window.
- All of the question answering fields in LumiNUS Quiz are paragraph-sized text entry forms. When answering MCQ and MRQs, just enter the appropriate letter(s) or “None” if an MRQ should have no options selected.
- Do remember that you will need to key in and submit your answers to the according assessment system as designated by your proctor or by exam central.

ANSWERS VERSION 1

1. *Evaluation Measures (MRQ; 4 options; choose all that apply; 4 marks)*. We know that the output of a binary classifier is the probability of the positive class. If the probability is larger or equal to 0.5, then we predict the positive class; less than 0.5, we predict a negative class.

   Say we change the threshold to 0.6. In other words, we now predict the postive class only when if the probability is larger or equal to 0.6.

   How will the **precision** and **recall** of the positive class change?

   (a) Precision rate increases or remains unchanged.
   (b) Precision rate decreases.
   (c) Recall rate increases.
   (d) Recall rate decreases or remains unchanged.

   Explanation: *The precision rate increases or remains unchanged, and Recall rate decreases or remains unchanged.*
   We will only the flip previously negatively predicted instances to positive class when the threshold increases. This results in fewer false positives. So the precision rate may only increase. However, we will predict a greater number of false negatives if we predict fewer positive class. So the recall rate may only decrease. Revise Lecture 04, segment on evaluating text classification.

[Questions 2–3] *Language Models (7 marks each; 14 marks total).* We would like to create a bigram language model from the following three simplified natural language sentences:

1. <s> barbara has car and truck </s>
2. <s> i have five red cars and barbara has five black trucks </s>
3. <s> i have a red car and barbara has five red cars </s>

Then the word counts of the above sentences are as follows:

No	token	count
1	</s>	3
2	<s>	3
3	a	1
4	and	3
5	barbara	3
6	black	1
7	car	2
8	cars	2
9	five	3
10	has	3
11	have	2
12	truck	1
13	trucks	1
14	i	2
15	red	3

With these statistics, please answer the following 2 questions. *Hint: you only need to calculate certain statistics to answer the questions.*
If we are using interpolated backoff for smoothing with $\delta = 0.75$, where

$$P_{IB}(w_n|w_{n-1}) = \{ \begin{matrix} \frac{C(w_{n-1}w_n)-\delta}{C(w_{n-1})} & \text{if } C(w_{n-1}w_n) \geq 1 \\ \alpha(w_{n-1}) \cdot P(w_n), & \text{if } C(w_{n-1}w_n) = 0, \end{matrix}$$

2. Calculate $P_{IB}(car|red)$. Show all work.

Explanation: *Revise Lecture 03, n-grams, backoff and interpretation, and Kneser-Ney smoothing topics. Here, we use the first case because* $C(redcar) >= 1$.
$P_{IB}(car|red) = (C(redcar) - \delta)/C(red)$
$P_{IB}(car|red) = (1 - 0.75)/3 = 0.083$

3. Calculate $P_{IB}(cars|red)$. Show all work.

Explanation: *Again, employ the first case as* $C(redcars) >= 1$.
$P_{IB}(cars|red) = (C(redcars) - \delta)/C(red)$
$P_{IB}(cars|red) = (2 - 0.75)/3 = 0.416$

[Questions 4–5] *Naïve Bayes (16 marks; 8 marks each)* A collection of reviews about comedy movies (data D) contains the following keywords and binary labels for whether each movie was *funny* (+) or *not funny* (−). The data are shown below; for example, the text of Review 1 contains 2 tokens of the word "laugh".

	laugh	hilarious	awesome	dull	yawn	bland	*Funny?*
Review 1	2	1	1	1	1	0	Yes
Review 2	3	1	2	0	0	1	Yes
Review 3	0	1	0	2	1	0	No
Review 4	2	1	1	4	2	2	No

4. Assume that you have trained a standard, non-smoothed Naïve Bayes model on data D to detect *funny* vs. *not funny* movie reviews. Compute the models predicted score for funny and not funny to the following sentence S (i.e., $P(+|S)$ and $P(-|S)$), and determine which label the model will apply to S. Show all work for full credit.

this film was hilarious i did nt yawn once

Explanation: We refer to Lecture 04's Naïve Bayes section for computing this solution.

(Tokenized) $S =$ ["this", "film", "was", "hilarious", "i", "didn't", "yawn", "once"]
$P(+|S) = P(+) \times P(hilarious|+) \times P(yawn|+)$
$P(-|S) = P(-) \times P(hilarious|-) \times P(yawn|-)$
$P(+) = \frac{2}{4} = 0.5$
$P(-) = \frac{2}{4} = 0.5$
$P(hilarious|+) = \frac{2}{13}$
$P(hilarious|-) = \frac{2}{16}$
$P(yawn|+) = \frac{1}{13}$
$P(yawn|+) = \frac{3}{16}$
So, $P(+|S) = 0.5 \times \frac{2}{13} \times \frac{1}{13} = 5.92 \times 10^{-3}$
And, $P(-|S) = 0.5 \times \frac{2}{16} \times \frac{3}{16} = 11.7 \times 10^{-3}$
Since $P(-|S) > P(+|S)$, the model labels S as -

5. Apply add-1 smoothing and recompute the Naïve Bayes models predicted scores for S. Did the label change?

Explanation: We refer to Lecture 04's Naïve Bayes section for computing this solution, where the second half of the runthrough illustrates the add-1 (Laplace) smoothing.

Recounting the counts of $P(word|+)$ and $P(word|-)$ from the previous question:
$P(hilarious|+) = \frac{2+1}{13+6}$
$P(hilarious|-) = \frac{2+1}{16+6}$
$P(yawn|+) = \frac{1+1}{13+6}$
$P(yawn|+) = \frac{3+1}{16+6}$
So, $P(+|S) = 0.5 \times \frac{3}{19} \times \frac{2}{19} = 8.31 \times 10^{-3}$
And, $P(-|S) = 0.5 \times \frac{3}{22} \times \frac{4}{22} = 12.4 \times 10^{-3}$
Since $P(-|S) > P(+|S)$, the model labels S as -. The label did not change.

[Questions 6–8] *Language Model True/False Questions (2 marks each; 6 marks total)*. For the following statements, type a "T" if the statement is true, or "F" otherwise. No justification is necessary.

6. For a given n-gram model, if we do not use a smoothing function, we can calculate the perplexity but the perplexity will be lower compared to a model employing smoothing.

Explanation: *False. Let's recall the formula of perplexity first. It contains $\frac{1}{q(w_i|w_{i-1})}$. If we do not use the smoothing function, then some of the probabilities will be zero. We will then encounter the error of division by zero when calculating perplexity. See Lecture 03 materials on evaluating n-gram models for reference.*

7. Given a training set consisting of types with low frequency, it is better to use a higher-order *n*-gram model.

   Explanation: *False. If the dataset contains many low frequency types, then we may not observe many of the corresponding higher-order n-grams, hence most of the counts of the n-grams will be zero. This is problematic, so it is better if we use lower-order n-gram model when dealing with low-frequencies in our dataset. Revise Lecture 03 on smoothing for reference.*

8. In an open vocabulary scenarios, we use smoothing functions to mitigate unknown word problems.

   Explanation: *True. Open vocabulary means that we do not constrain the vocabulary to a limited set, and hence there will often be unknown words. We use smoothing to assign a plausible but small probability values to account for such cases. OOV is a central problem in NLP, related to sparsity, and smoothing is a technique to combat it. See Lectures 02 and 03 for reference.*

[Questions 9–11] *Words MCQ/MRQ/True False (9 marks total)* If your interpretation of the question relies on particular assumptions, please state them.

9. *(True/False; 2 marks)* When we talk of *expressivity* of natural language utterances, we assume that the *pragmatic interpretations* of the utterances are also equivalent.

   Explanation: *False. Expressivity means that there is more than one way that to state an utterance with the same meaning. However with different statements, the pragmatic contextual interpretation can be different; Reference Slide 43 in Lecture 01: "Please be quiet. The talk will begin shortly." and "Shut up! The talk is starting!" convey the same semantics but the implication of politeness differs between the two utterances.*

10. *(MCQ; 4 marks)* Assume two edit distance calculations: Edit Distance A with unit costs for insertion, deletion operations; and Edit Distance B with unit costs for insertion, deletion and substitution operations. Is it possible for substrings of either the target or the source to have *larger* edit distance than the original target and source string pair?

    (a) No to both.
    (b) Yes to both.
    (c) Yes only to Edit Distance A.
    (d) Yes only to Edit Distance B.

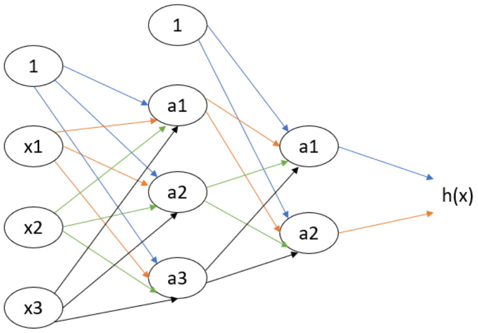
    Explanation: *Yes to both. For example on Slide 84 of Lecture 02, which illustrates Edit Distance A, we can see that the string pair (INT,EXECU) has optimal cost 8 but the string pair (INT, EXECUT) has cost 7. This is true even with substitutions have unit cost. Take the string pairs (INTE, EX) with cost 4 and (INTE, EXE) with cost 3. This is because the final letter in both longer cases align and give suboptimal structure with the shorter respective pairs. Reference Lecture 02, edit distance topic.*

11. *(MRQ with 5 options; 3 marks)* Which tokens below match the regex `^.[0-9]?$` ?

    (a) a0
    (b) .
    (c) !
    (d) b
    (e) $3

    Explanation: *This question was typeset poorly so we accept the answer where the "." (period) was not visible. In the case where the period was correctly visible, Options "b" and "a0" are correct. Where the period was not visible, all of the answers are not correct. Reference Lecture 02 first topic on regular expressions for revision.*

12. *Neural Networks (MCQ; 8 marks)*. Suppose we are using a neural network with **an input vector of length 3, one hidden layer with three neurons and two output neurons.** Additionally, the hidden neurons and the input will **include a bias.** We use **tanh function** ($tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$) as the nonlinearity. Heres the basic structure:



Suppose there is a data input **x** = $(4, 2, 1)$ and the weights for the network are (the first column is the weights of the bias vector $x_0 = 1$):

$$\mathbf{\Theta}^{[1]} = \begin{bmatrix} 0.2 & -0.5 & 0.3 & 0.4 \\ 0.5 & 0.2 & -0.2 & 0.1 \\ 0.3 & 0.3 & -0.1 & -0.3 \end{bmatrix}, \mathbf{\Theta}^{[2]} = \begin{bmatrix} -0.2 & 0.7 & -0.2 & 0.4 \\ 0.4 & 0.1 & 0.6 & -0.3 \end{bmatrix}$$

Calculate the output $a^{[2]} = (a_1, a_2)$ of the following neural network implementation (rounded to one decimal place).

(a) $a^{[2]}$ = (-0.5, 0.5)
(b) $a^{[2]}$ = (-0.5, -0.5)
(c) $a^{[2]}$ = (0.5, -0.5)
(d) $a^{[2]}$ = (0.5, 0.5)

Explanation: *$a^{[2]}$ = (-0.5, 0.5). Refer to Week 05, neural network segment for the forward propagation method. In the forward propogation stage, the values are as follows:*
$a_1^{[1]} = -0.66, a_2^{[1]} = 0.76, a_3^{[1]} = 0.76$
$a_1^{[2]} = tanh(-0.66 * 0.7 + 0.76 * -0.2 + 0.76 * 0.4 - 0.2) = -0.5$
$a_2^{[2]} = tanh(-0.66 * 0.1 + 0.76 * 0.6 + 0.76 * -0.3 - 0.4) = 0.5$
Partial marks may be awarded by staff for incomplete computations, staff decisions are final.

13. *TF·IDF (MCQ; 3 marks)*. In a particular document d containing T words in total, the word "language" appears k times. If in the whole collection with m documents, the word "language" appears $m/3$ times, then the $tf \cdot idf_{(\text{language},d)}$ is:

(a) $(log(k/T) + 1) \times log(m/3)$
(b) $(log(k/T) + 1) \times log(3)$
(c) $(log(k) + 1) \times log(m/3)$
(d) $(log(k) + 1) \times log(3)$

Explanation: *$(log(k) + 1) \times log(3)$, where the $(log(k) + 1)$ is the tf term, and the $log(3)$ is the idf term. Revise Lecture 04, Section tf–idf.*

This marks the end of this part of the exam.
These is no additional material beyond this point.