考古

第一題

- (a) `^[a-z]*b$`
- (b) `^[0-9]+.*[a-zA-Z]$`

1. **(10 pt)** Please write *regular expressions* for the following languages:
   (a) the set of all lower case alphabetic strings ending in *b*. **(5 pt)**
   (b) all strings that start at the beginning of the line with an integer and that end at the end of the line with a word. **(5 pt)**

字元	說明	例子
.	與任何單一字元比對 （字母、數字或符號）	.ool : cool 、fool 、 tool too. : tool 、took goo.gle : gooogle 、goodgle 、goo7gle
?	比對前字元0次或1次	be?d: bed 、be
*	比對前字元0次或多次	bag01* : bag0 、bag01 、bag011 、bag0111
+	比對前字元1次或多次	bag01+ : bad01 、bag011 、bag0111 、bag01111
|	或（不可在運算式最尾端）	facebook | instagram : 出現facebook或instagram black | jacket : 只要是黑色的商品或是外套類的商品
.*	.代表任意字元，*零次或多次比對 →比對所有可能的條件	

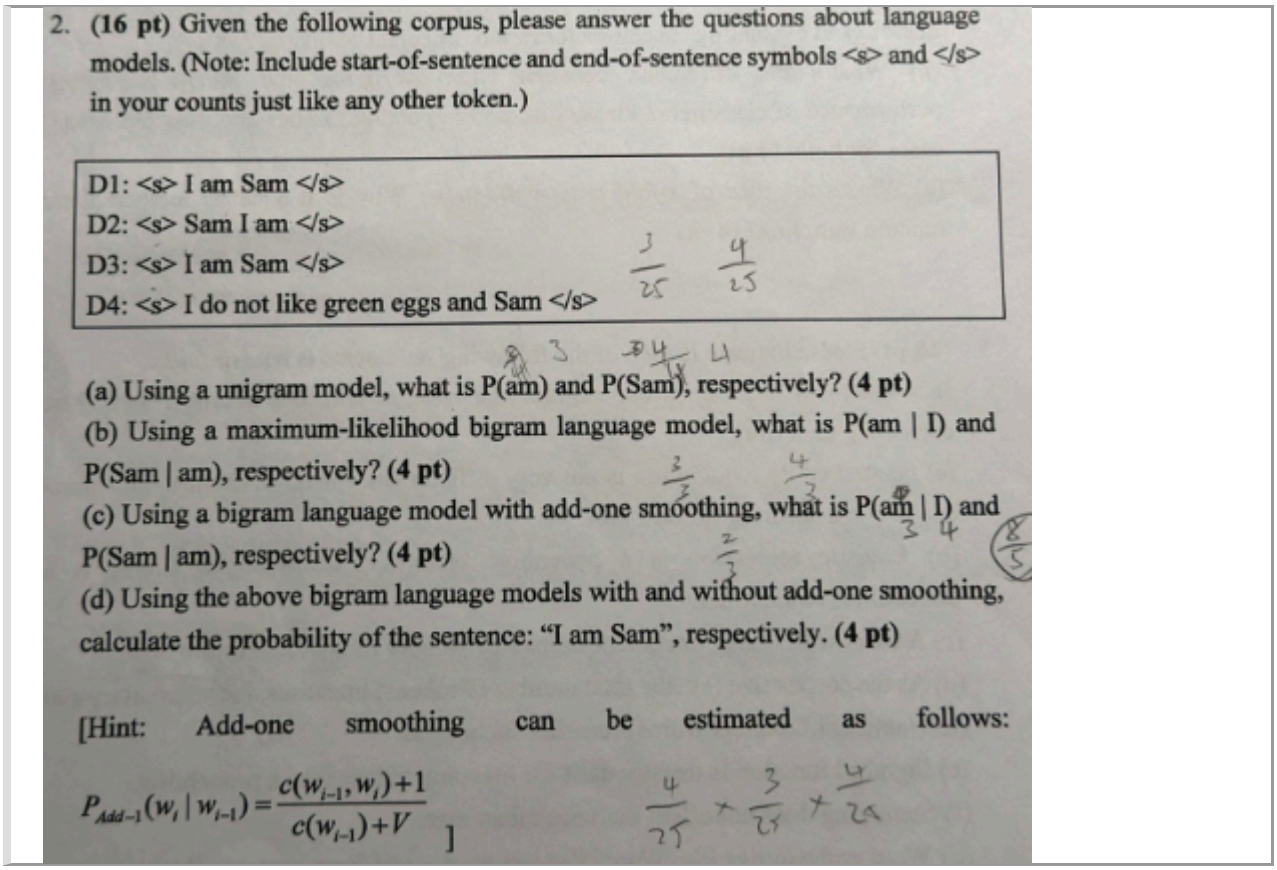
其他	說明	例子
^	比對開頭與符號鄰接字元相符的字串	^ap: ape 、app ; 無法完成比對 : tap 、cap
$	比對結尾語符號鄰接字元一致的字串	ap$: leap 、rap ; 無法完成比對 : app 、ape
\	表示鄰接字元應視為常值， 而非運算式中的字元	\. : 相鄰的原點應被視為句點或小數點 192\.168.\.38\.42 – 比對 ip 192.168.38.42

分組符號	說明	例子
( )	比對所有跟括號內字元 排列順序完全相符的字串	(ele) : elephant 、telephone (thank): thanks 、thankyou 、thankful grand(pa | ma): grandpa 、grandma
[ ]	以任一順序比對括號中的字元與字串	[abc] : a 、b 、c 、ab 、ac 、bc 、ba 、ca 、 cb 、abc 、acb 、bca 、bac 、cab 、cba [10] : 012 、124 、150 、310 、079
-	根據括號中的字元範圍 比對字串中的任一部分	[0-9] : 比對0到9之間的任一數字 [A-E] : A 、B 、C 、D 、E red[1-3]: red1 、red2 、red3 blue15[3-5]: blue153 、blue154 、blue155

- [^] 將方括號內的^放在開頭,表示不匹配方括號內的任何字符
- [0-9A-Za-z] 多個範圍可以組合,這裡表示所有數字和字母
- {n} 匹配前面字符剛好n次
- {n,} 匹配前面字符至少n次
- {n,m} 匹配前面字符n到m次
- \s 匹配任何空白字符(空格、制表符、換頁符等)
- \S 匹配任何非空白字符

第二題

- (a)
  - P(am)： 出現am的次數 / 總字數(包含<s>和</s>) = 3/25
  - P(Sam)： 4/25
- (b)
  - P(am|I)： 出現 I am的機率 / I的機率 = 3/4
  - P(Sam|am)： 2/3
- (c)
  - P(am|I)： (出現 I am 的機率 + 1) / [I 的機率 + 出現的詞彙量(不重複)] = 4/15
  - P(Sam|am)： 3/14
- (d)
  - P(I|<s>) * P(am|I) * P(Sam|am) * P(</s>|Sam)
  - with add-one： (4/15)*(4/15)*(3/14)*(4/15) = 0.00406 = 192/47250
  - without add-one： (3/4)*(3/4)*(2/3)*(3/4) = 9/32



2. **(16 pt)** Given the following corpus, please answer the questions about language models. (Note: Include start-of-sentence and end-of-sentence symbols <s> and </s> in your counts just like any other token.)

D1: <s> I am Sam </s>
D2: <s> Sam I am </s>
D3: <s> I am Sam </s>
D4: <s> I do not like green eggs and Sam </s>

(a) Using a unigram model, what is P(am) and P(Sam), respectively? **(4 pt)**
(b) Using a maximum-likelihood bigram language model, what is P(am | I) and P(Sam | am), respectively? **(4 pt)**
(c) Using a bigram language model with add-one smoothing, what is P(am | I) and P(Sam | am), respectively? **(4 pt)**
(d) Using the above bigram language models with and without add-one smoothing, calculate the probability of the sentence: "I am Sam", respectively. **(4 pt)**

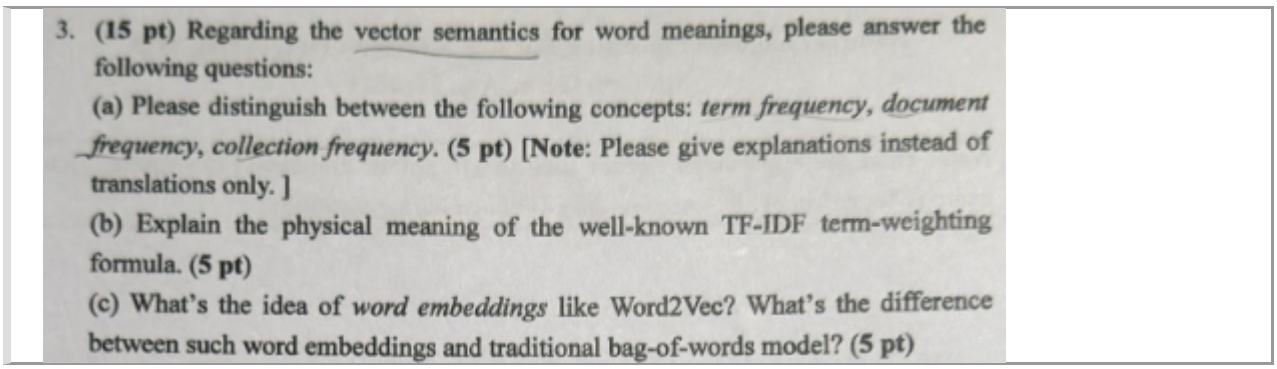
[Hint: Add-one smoothing can be estimated as follows:

$$P_{Add-1}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)+1}{c(w_{i-1})+V}$$

]

第三題

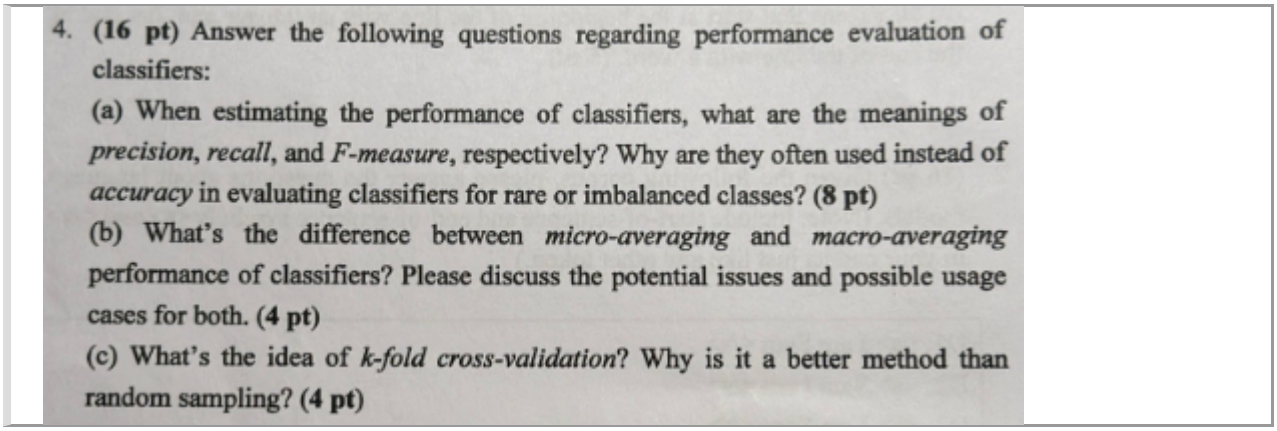
- (a)
  - term frequency: 一個詞在單個文件中出現的次數
  - document frequency: 一個詞在整個文件集合中出現的文件數
  - collection frequency: 一個詞在整個文件集合中出現的總次數
- (b)
  - TF-IDF是一種常用的詞彙加權方法。TF (term frequency) 表示詞頻,而IDF (inverse document frequency) 表示文件出現頻率的反向,用於降低常見詞的權重。這種加權方式能夠突出重要的詞並降低無關詞的影響。
- ©
  - Word embeddings 如Word2Vec,是將單詞映射到連續向量空間的技術,能捕捉詞彙的語義和上下文關係。傳統的bag-of-words模型將文檔表示為獨熱向量,無法體現詞之間的關係。



3. **(15 pt)** Regarding the vector semantics for word meanings, please answer the following questions:
(a) Please distinguish between the following concepts: *term frequency*, *document frequency*, *collection frequency*. **(5 pt)** [**Note**: Please give explanations instead of translations only.]
(b) Explain the physical meaning of the well-known TF-IDF term-weighting formula. **(5 pt)**
(c) What's the idea of *word embeddings* like Word2Vec? What's the difference between such word embeddings and traditional bag-of-words model? **(5 pt)**

第四題

- (a)
  - 準確率是正確預測的比例
  - 召回率是實際正樣本中被正確識別的比例
  - F-measure結合了準確率和召回率
  - 在數據不平衡的情況下,上述指標比單純準確率更有意義。
- (b)
  - Micro-averaging將所有樣本視為一個大樣本,不考慮類別差異
  - Macro-averaging先對每個類別計算各指標,再進行平均
  - Macro-averaging更適合不平衡數據,而micro-averaging對頻繁類別更敏感。
- ©
  - K-fold交叉驗證將數據分為K份,每次使用K-1份作為訓練集,剩下1份作為測試集,經過K次後取平均結果。相比隨機抽樣,它利用了所有數據,結果更可靠。

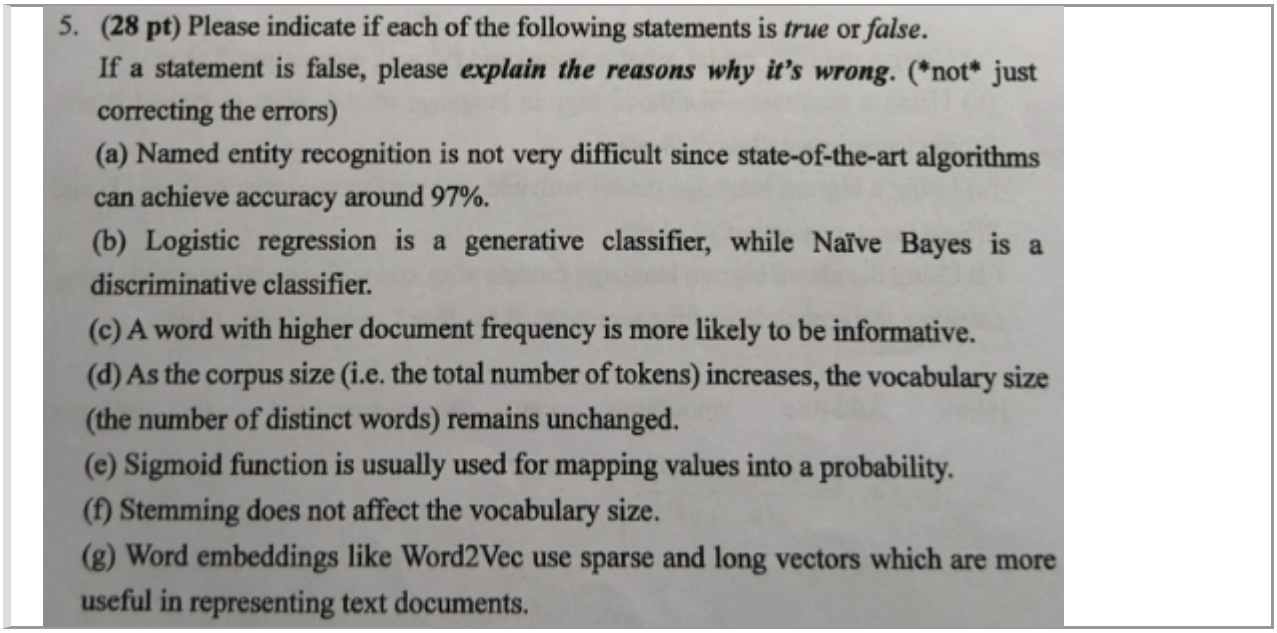


4. **(16 pt)** Answer the following questions regarding performance evaluation of classifiers:
(a) When estimating the performance of classifiers, what are the meanings of *precision*, *recall*, and *F-measure*, respectively? Why are they often used instead of *accuracy* in evaluating classifiers for rare or imbalanced classes? **(8 pt)**
(b) What's the difference between *micro-averaging* and *macro-averaging* performance of classifiers? Please discuss the potential issues and possible usage cases for both. **(4 pt)**
(c) What's the idea of *k-fold cross-validation*? Why is it a better method than random sampling? **(4 pt)**

第五題

- (a)
  - 錯誤。由於模糊性、上下文依賴性和實體類型的多樣性等因素，任務本身可能相當具有挑戰性。高精度通常是複雜模型和大量計算資源的結果。
- (b)
  - 錯誤。Logistic regression是一種判別式分類器,而Naive Bayes則是生成式分類器。
- ©
  - 錯誤。高文件頻率的詞往往是常用詞,反而不太含信息量。
- (d)
  - 錯誤。隨著token總數增加,詞彙量通常也會持續增長,而不是保持不變。
- (e)
  - 正確。Sigmoid函數常用於將值映射到概率範圍(0,1)。
- (f)

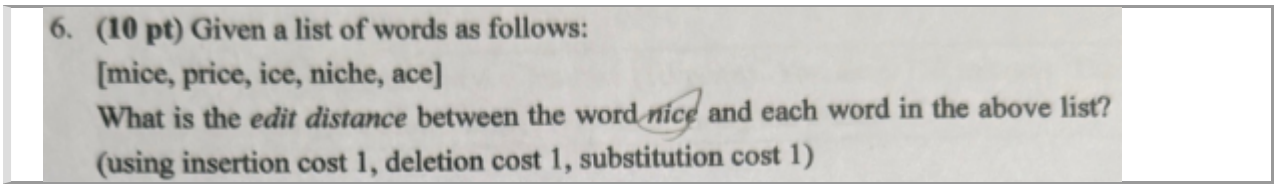
    - 錯誤。詞幹提取將單字簡化為其詞幹或詞根形式，這可以透過將單字的相似形式合併為單一術語來顯著減少詞彙量。
- (g)
    - 錯誤。Word embeddings產生的是dense vector,而非sparse vector。



5. **(28 pt)** Please indicate if each of the following statements is *true* or *false*.
If a statement is false, please ***explain the reasons why it's wrong***. (*not* just correcting the errors)
(a) Named entity recognition is not very difficult since state-of-the-art algorithms can achieve accuracy around 97%.
(b) Logistic regression is a generative classifier, while Naïve Bayes is a discriminative classifier.
(c) A word with higher document frequency is more likely to be informative.
(d) As the corpus size (i.e. the total number of tokens) increases, the vocabulary size (the number of distinct words) remains unchanged.
(e) Sigmoid function is usually used for mapping values into a probability.
(f) Stemming does not affect the vocabulary size.
(g) Word embeddings like Word2Vec use sparse and long vectors which are more useful in representing text documents.

第六題

- 計算單詞"nice"與列表中每個單詞的編輯距離(插入、刪除和替換的代價均為1):
- mice - 1 (替換n為m)
- price - 2 (插入p, 替換n為r)
- ice - 1 (刪除n)
- niche - 1 (插入h)
- ace - 2 (刪除n, 替換i為a)



6. **(10 pt)** Given a list of words as follows:
[mice, price, ice, niche, ace]
What is the *edit distance* between the word *nice* and each word in the above list?
(using insertion cost 1, deletion cost 1, substitution cost 1)

第七題

第一步:建立詞彙集合

- 詞彙集合 = {fun, couple, love, fast, furious, shoot, fly}
- 詞彙數量 |V| = 7

第二步:計算每個類別的先驗概率

- P(comedy) = 2/5

- P(action) = 3/5

第三步:使用加法平滑計算條件概率

- 加法平滑參數設為1
- 只需計算 new review 的機率

對於comedy類別:

- P(couple|comedy) = (2+1) / (9+7) = 3/16
- P(fast|comedy) = (1+1) / (9+7) = 2/16
- P(shoot|comedy) = (0+1) / (9+7) = 1/16
- P(fly|comedy) = (1+1) / (9+7) = 2/16

對於action類別:

- P(couple|action) = (0+1) / (11+7) = 1/18
- P(fast|action) = (2+1) / (11+7) = 3/18
- P(shoot|action) = (4+1) / (11+7) = 5/18
- P(fly|action) = (1+1) / (11+7) = 2/18

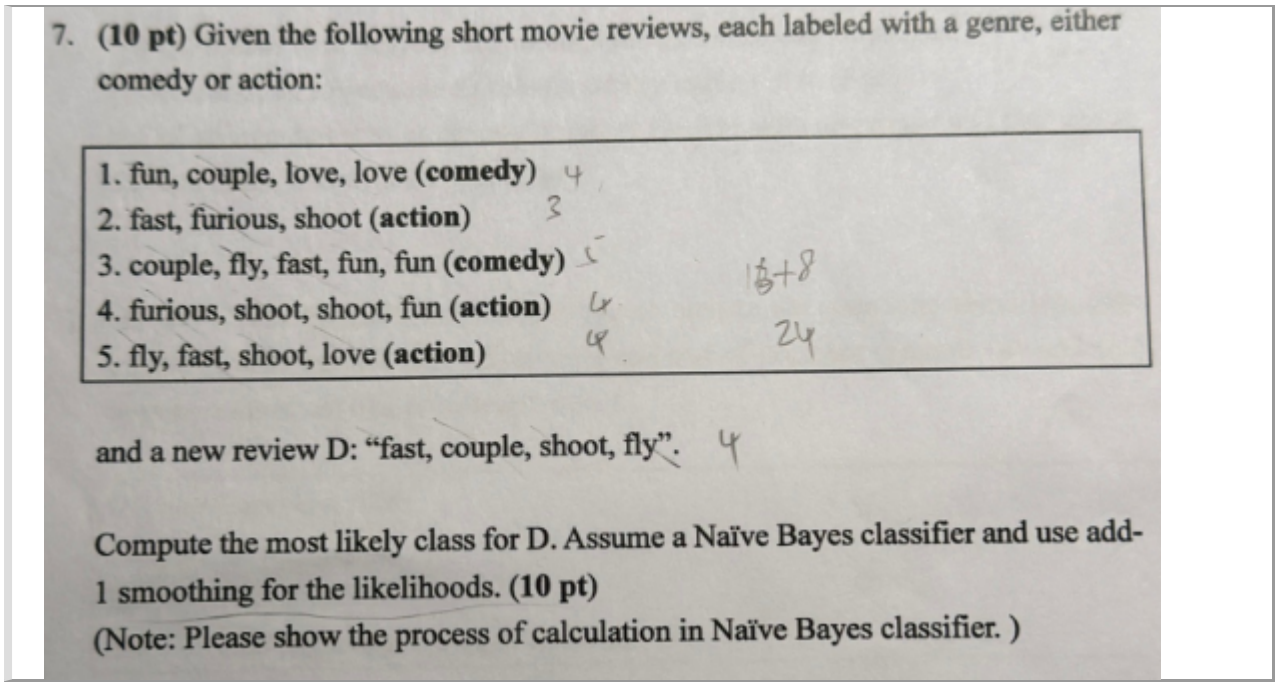
第四步:對新評論"fast, couple, shoot, fly"計算在每個類別下的概率

For comedy:

- P(comedy) * P(fast|comedy) * P(couple|comedy) * P(shoot|comedy) * P(fly|comedy)
- = (2/5) * (2/16) * (3/16) * (1/16) * (2/16)
- = 0.0000732421875

For action:

- P(action) * P(fast|action) * P(couple|action) * P(shoot|action) * P(fly|action)
- = (3/5) * (3/18) * (1/18) * (5/18) * (2/18)
- = 1.7146776406035665294924554183813e-4
- 由於P(action) > P(comedy),所以最可能的類別是action。



7. **(10 pt)** Given the following short movie reviews, each labeled with a genre, either comedy or action:

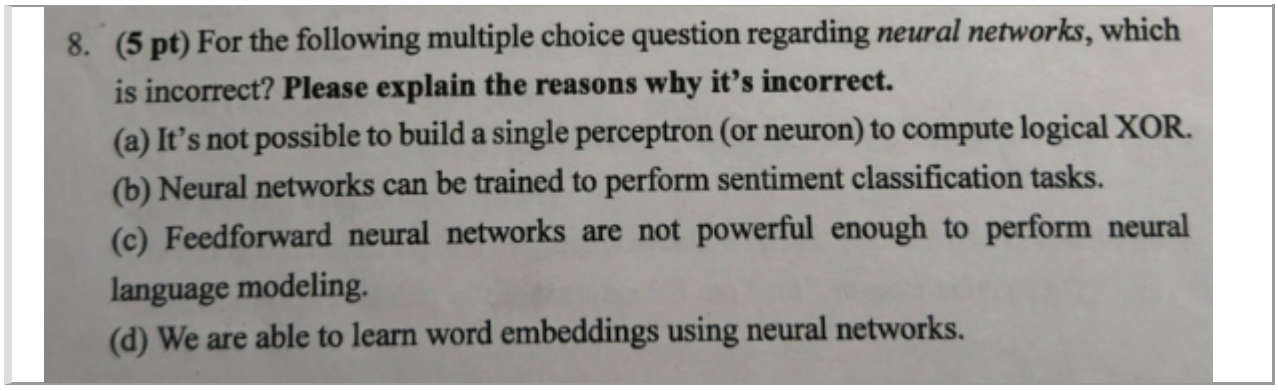
1. fun, couple, love, love **(comedy)**
2. fast, furious, shoot **(action)**
3. couple, fly, fast, fun, fun **(comedy)**
4. furious, shoot, shoot, fun **(action)**
5. fly, fast, shoot, love **(action)**

and a new review D: "fast, couple, shoot, fly".

Compute the most likely class for D. Assume a Naïve Bayes classifier and use add-1 smoothing for the likelihoods. **(10 pt)**
(Note: Please show the process of calculation in Naïve Bayes classifier.)

第八題

- (a)正確。單個感知器無法計算邏輯XOR運算。
  - 感知器是一種簡單的線性分類器,它通過將輸入向量與權重向量相乘,然後與一個偏置項相加,最後通過激活函數(如階跡函數)得到輸出。
  - XOR的輸出無法用一條直線將其完全分開(即無法找到一個權重向量使所有正例在一側,所有反例在另一側)。
- (b)正確。神經網絡可以被訓練執行情感分類等任務。
- ©錯誤。前饋神經網絡已展現出在自然語言處理等任務上的強大能力。
  - 前饋神經網路(feedforward neural networks)是一種基本的人工神經網路架構,其中資訊單向傳遞從輸入層到隱藏層,再到輸出層,無回路或循環連接。儘管是一種相對簡單的架構,但前饋神經網路已被證明對於許多自然語言處理任務(如神經語言模型)具有強大的建模能力。許多知名語言模型如word2vec、BERT等都是基於前饋神經網路的變種架構。
- (d)正確。神經網絡可用於學習詞嵌入向量表示。



8. **(5 pt)** For the following multiple choice question regarding *neural networks*, which is incorrect? **Please explain the reasons why it's incorrect.**
(a) It's not possible to build a single perceptron (or neuron) to compute logical XOR.
(b) Neural networks can be trained to perform sentiment classification tasks.
(c) Feedforward neural networks are not powerful enough to perform neural language modeling.
(d) We are able to learn word embeddings using neural networks.

Chapter 2

Words和語料庫

術語

- Token: 文本中的基本單位,如單詞或標點符號
- Type: 語料庫中的詞彙單元
- Wordform: 單詞的全稱

Heaps定律

- V = k * N^β
  - V: 語料庫中的詞彙量
  - N: 語料庫的Token數量
  - k和β是常數,通常0.67 < β < 0.75
  - 描述詞彙量與Token數量的統計規律關係

語料庫注意事項

- 不同語言、變種、體裁、作者人口統計學等都會導致詞彙差異
- 使用語料庫應提供詳細的"語料庫簡介"(Corpus Datasheets)

標記化 (Tokenization)

空格分詞 (Space-based Tokenization)

- 適用於使用空格作為單詞分隔符的語言
- 可使用Unix工具進行標記化和計數

單字符分詞

- 常見於沒有明顯詞界的語言,如中文、日語等,將每個字符視為一個Token

子詞標記(Subword Tokenization)

- 使用基於數據的算法學習如何切分Token,包括單詞和子詞
- 都包含兩個部分:
  - Token學習器:從原始語料庫中學習一個Token集合(詞彙表)
  - Token分割器:根據學習到的詞彙表對測試句子進行標記化

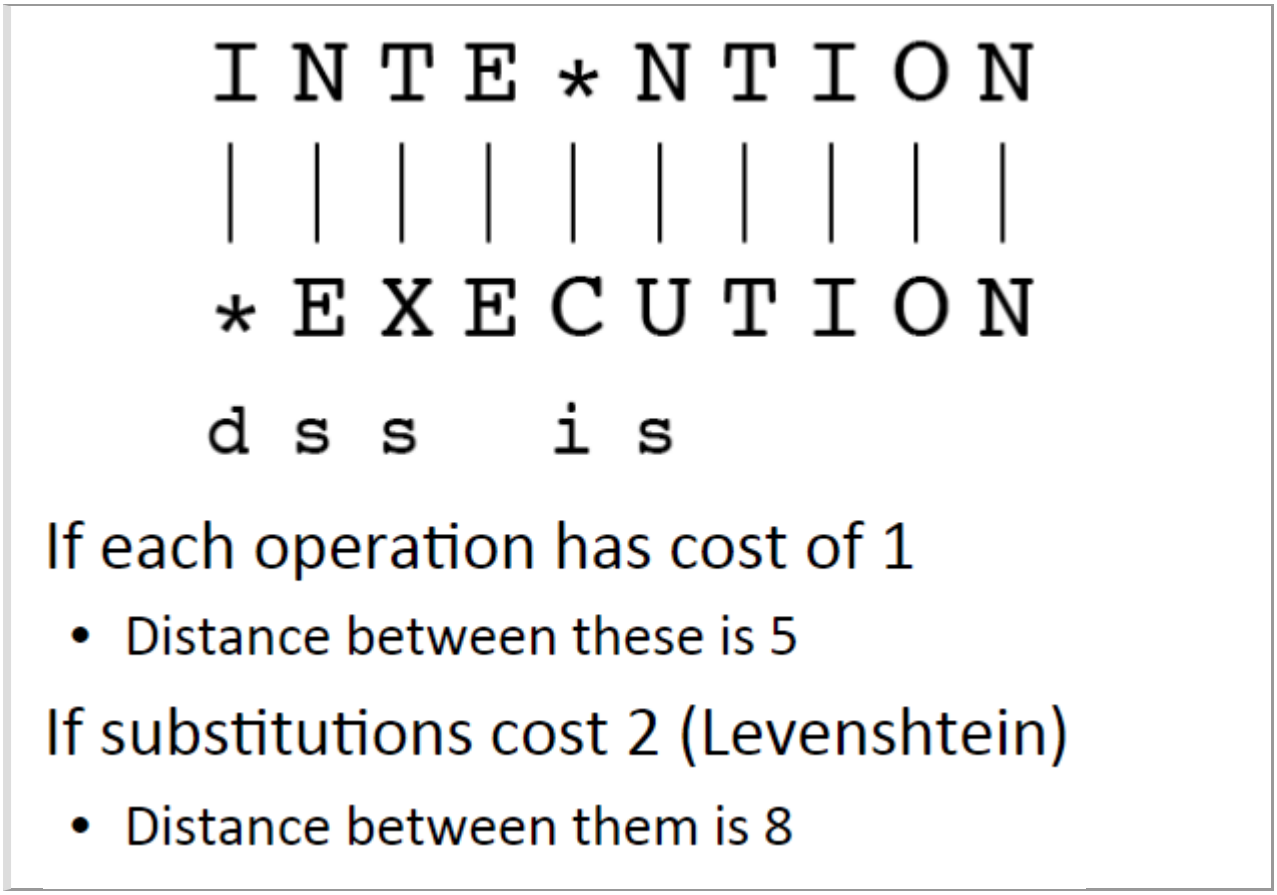
Byte Pair Encoding (BPE)

- Token學習器:
  - 初始詞彙表為所有單個字符

    - 重複以下步驟直到達到指定的合併次數:
        1. 在語料庫中找到頻率最高的相鄰字符對(如'A','B')
        2. 將該字符對合並為一個新Token'AB'加入詞彙表
        3. 在語料庫中將所有'AB'替換為新Token'AB'
- Token分割器:
    - 貪婪地按學習到的合併順序對測試句子進行分割
- 優點 : 可以很有效地平衡词典大小和编码步骤数
- 缺點 : 对于同一个句子，可能会有不同的 Subword 序列。不同的 Subword 序列会产生完全不同的 id 序列表示

edit distance table

- 在自然語言處理(NLP)中,edit distance table指的是用來計算兩個字串之間編輯距離的動態規劃表。編輯距離是指將一個字串轉換成另一個字串所需的最少編輯操作次數,通常包括插入、刪除和替換操作。
- Edit distance table是編輯距離計算的核心。它是一個二維矩陣,其中行列分別對應兩個待比較的字串。表格中的每個元素表示將源字串的前i個字元轉換為目標字串的前j個字元所需的最小編輯距離。通過自底向上的動態規劃方式,可以逐步填充編輯距離表,最終得到兩個字串之間的編輯距離。
- Time : O(nm)
- Space : O(nm)
- Backtrace : O(n+m)



Chapter 4

naive bayes

- 在自然語言處理(NLP)中,樸素貝葉斯(Naive Bayes)分類器是一種簡單而有效的機率分類模型,常被用於文字分類、垃圾郵件過濾、情緒分析等任務。
- 樸素貝葉斯分類器的"樸素"假設是:在給定目標類別的情況下,預測特徵之間是相互獨立的。 即使這個假設在實際情況下可能不太符合,但由於演算法本身的簡單性和高效性,樸素貝葉斯在NLP中仍有廣泛應用。
- 具體來說,樸素貝葉斯分類器包括以下幾個主要步驟:
- 文字表示
  - 將文字轉換為詞條頻率向量(如詞袋模型BOW),每個維度對應一個詞條在文件中出現的頻率。
- 機率估計
  - 基於訓練資料,估計每個類別c下不同特徵詞條x的條件機率P(x|c)。 常用的估計方法有加示數平滑、Laplace平滑等。
- 貝葉斯公式
  - 利用貝葉斯定理,計算在給定一個文檔x的條件下,它屬於類別c的後驗機率:
  - `P(c|x) = P(x|c)P(c) / P(x)`
- 分類決策
  - 對於一個待分類文件,計算它屬於每個類別的後驗機率,將其歸於後驗機率最大的那一類。
- 樸素貝葉斯的優點是簡單、有效率、對缺失資料較不敏感。 缺點是"樸素"假設在實際中常常不成立,無法有效捕捉特徵之間的關聯。
- 儘管簡單,但在一些大規模資料集上,樸素貝葉斯分類器的性能仍然可以與其他複雜模型相當。 它被廣泛應用於垃圾郵件、新聞分類等傳統NLP任務,也常作為強基線對深度學習模型進行對比。 總的來說,樸素貝葉斯是一種非常實用且成熟的NLP分類模型。

Chapter 5

logistic regression

- 在自然語言處理(NLP)中,Logistic Regression(logistic回歸)是一種廣泛應用的機器學習算法,通常用於文本分類、情感分析等任務。
- Logistic Regression的主要思想是,將文本映射到一個介於0和1之間的值,表示該文本屬於目標類別的概率。具體來說:
  - 文本表示

        - 首先將文本轉換為特徵向量,如TF-IDF、Word Embedding等。每個維度表示一個特徵在該文本中的權重。
    - 線性組合
        - 將特徵向量與權重向量做線性組合,得到一個標量值z = w^T * x。其中w為權重向量,x為特徵向量。
    - Logistic函數
        - 將線性組合的結果z通過Logistic函數(Sigmoid函數)映射,得到介於0和1之間的值y = 1 / (1 + exp(-z))。
    - 概率解釋
        - 將y解釋為文本屬於目標類別的概率。若y>0.5,則判定為目標類別,否則為非目標類別。
    - 模型訓練
        - 使用訓練數據,通過最大似然估計或者最小化交叉熵等目標函數,學習最優權重向量w。
- Logistic Regression的優點是簡單、高效且易於理解。它假設特徵與目標類別的對數odds比值呈線性關係。當然,對於複雜的NLP任務,單層的Logistic回歸表現往往不如深度神經網絡。但由於其解釋性強,在一些關鍵場景下仍有不可替代的作用。
- 總之,Logistic Regression是NLP領域一種基礎且重要的分類模型,通過概率框架將文本映射到類別空間。許多更高級的神經網絡模型中,也會使用Logistic作為輸出層的激活函數。

Chapter 6

embadding

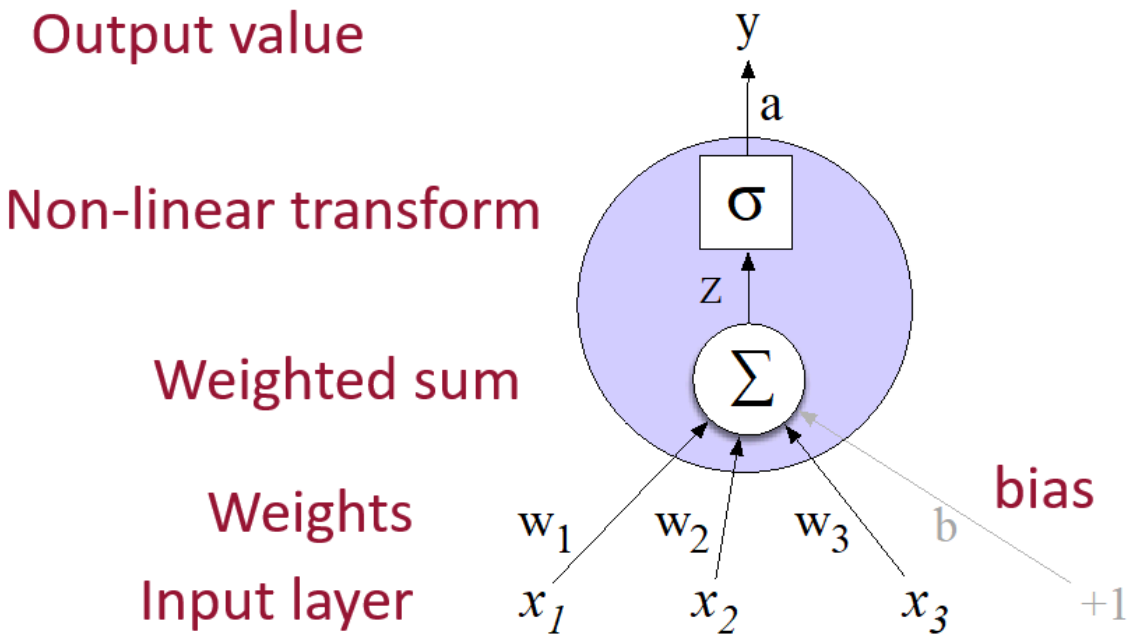
- 在自然語言處理(NLP)中,embedding 指的是將離散的符號(例如單詞)映射到連續的向量空間中的向量表示。這種向量表示能夠捕捉單詞之間的語義關係和上下文信息。embedding 是許多現代 NLP 模型和技術的基礎,例如 word2vec、GloVe、BERT 等。
- 具體來說,embedding 主要有以下幾個作用:
    - 降維
        - 單詞通常使用一個熱編碼的高維稀疏向量表示,embedding 則將其投影到一個低維且密集的向量空間,這樣可以大大降低計算和存儲的開銷。
    - 揭示語義關係
        - 在 embedding 向量空間中,語義相似的單詞會被映射到彼此較近的位置。例如"國王"和"女王"的向量很接近。這種向量表示揭示了單詞之間的語義聯繫。
    - 共現關係
        - 語言中共現頻繁的單詞會被映射到相近的向量。embedding以分佈式表示捕獲了單詞在語料庫中的上下文分佈信息。
    - 傳遞性

    - embedding空間保留了單詞之間的語義組合關係。例如 vec("國王") - vec("男人") + vec("女人") 會得到一個與 vec("女王")非常接近的向量。
- 總之,embedding 是將離散符號映射到連續向量表示的一種技術,使得單詞可以用單一向量體現其語義和上下文信息,這對NLP任務至關重要。大多數現代NLP模型和算法都基於embedding作為單詞的基本表示形式。

Chapter 7

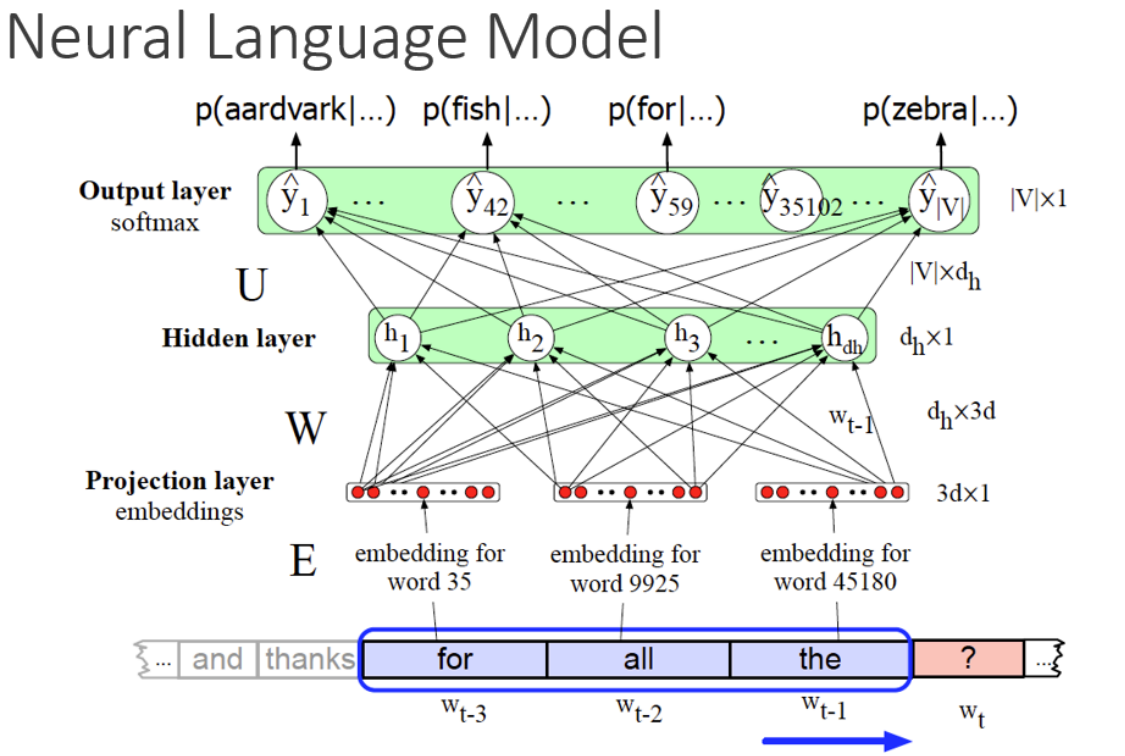
NN

- 第一張圖展示了一個單一神經元(感知器)的運作原理。它包含以下幾個主要部分:
    - 輸入層(Input layer)包含多個輸入變數x1、x2、x3等,分別乘以對應的權重w1、w2、w3。
    - 加權和(Weighted Sum)將所有輸入值乘以相應權重後相加,並加上一個偏置值b。
    - 非線性轉換(Non-linear Transform)通過激活函數σ(常見如Sigmoid或ReLU)對加權和的結果進行非線性轉換,產生輸出值a。
    - 輸出值(Output Value)即為神經元的最終輸出y。
- 這個簡單的神經元模型是構建更複雜神經網絡的基礎單元。通過組合多個神經元並引入多層結構,可以實現對複雜數據的建模和預測。



- 描繪了一種神經語言模型(Neural Language Model)的架構。這種模型廣泛應用於自然語言處理任務中,如機器翻譯、文本生成等。它的主要組成部分包括:
    - 嵌入層(Projection Layer)將每個單詞映射為一個固定長度的向量表示(embedding)。
    - 隱藏層(Hidden Layer)對輸入的單詞嵌入進行非線性轉換,捕獲單詞之間的上下文關係。

  - 輸出層(Output Layer)使用softmax函數計算給定上下文中每個可能單詞的概率分佈。
  - 模型的輸入是一個單詞序列,輸出則是下一個單詞的概率分佈。
- 通過訓練這種模型在大量文本數據上,它可以學習到語言的統計規律,並用於生成新的文本、完成句子等自然語言生成任務。



Chapter 8

POS tagging

- 詞性標註是將句子中的每個單詞賦予相應的詞性標記,如名詞(NN)、動詞(VB)、形容詞(JJ)等。這項任務對於許多NLP應用程序至關重要,如句法分析、機器翻譯、信息提取等。
- 例句: The/DT young/JJ cat/NN sat/VBD on/IN the/DT mat/NN.
- 這個句子中,每個單詞都被標註了其詞性,如DT代表限定詞(the)、JJ代表形容詞(young)、NN代表名詞(cat、mat)等。

Markov Model

- 馬爾可夫模型是一種用於建模序列數據的統計模型。它基於馬爾可夫假設:未來狀態的條件概率分佈僅依賴於當前狀態,而與過去的狀態無關。
- 在NLP中,隱馬爾可夫模型(Hidden Markov Model, HMM)是一種應用廣泛的馬爾可夫模型。它由隱藏的馬爾可夫鏈(隱狀態序列)和可觀察到的輸出序列組成,常用於詞性標註、命名實體識別、音素到文本的轉錄等任務。

- 例如計算一個句子"I am a student"中每個單詞出現的概率:
- P(I|START) = 0.4
- P(am|I) = 0.3
- P(a|am) = 0.5
- P(student|a) = 0.2

Named Entity Recognition (NER)

- NER是從非結構化文本中識別出實體名稱(如人名、地名、組織名等)並對它們進行分類的任務。準確的NER對於諸如信息抽取、問答系統、關係抽取等應用程序非常重要。
- Julia Roberts is an American actress.
- 輸出: Julia/PERSON Roberts/PERSON is an American/NATIONALITY actress/NOUN

BIO Tagging

- BIO標註是一種常用的序列標註策略,主要應用於命名實體識別等任務。它將單詞標註為B(實體開始)、I(實體中間)或O(非實體)。
- 具體來說,對於一個實體,第一個單詞標註為B,剩餘單詞標註為I,非實體單詞標註為O。這種標註方式將實體內部單詞與實體邊界分開,從而方便模型學習實體類型和範圍。
- BIO標註可以根據實際需求擴展,如BIOES標註(分別標註單獨實體、實體開始、實體中間、實體結尾和單獨非實體)。
- Harry Potter is a series of fantasy novels.
- 輸出: Harry/B-PERSON Potter/I-PERSON is/O a/O series/O of/O fantasy/O novels/O