IMPERIAL COLLEGE LONDON

TIMED REMOTE ASSESSMENTS 2020-2021

MEng Honours Degree in Mathematics and Computer Science Part IV
MEng Honours Degrees in Computing Part IV
MSc Advanced Computing
MSc Artificial Intelligence
MSc in Computing (Specialism)
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant assessments for the Associateship of the City and Guilds of London Institute

PAPER COMP70016=COMP97115=COMP97116

NATURAL LANGUAGE PROCESSING

Friday 19 March 2021, 10:00
Duration: 140 minutes
Includes 20 minutes for access and submission

Answer ALL FOUR questions
Open book assessment

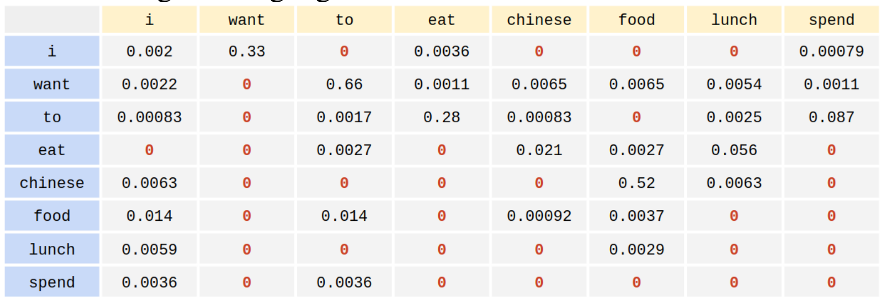
Paper contains 4 questions

1 a Consider training the skip-gram model of word2vec with the following settings: BPE vocabulary = 50,000, word embedding dimensionality (i.e. hidden layer dimensionality) = 300, context window = 5.

i) In the softmax variant, for a single training sample, how many neurons need to be updated in the input weight matrix?

ii) In the negative sampling variant, the hyperparameter k (number of negative samples) is generally decided as a function of the size of the training corpus for word2vec. Explain why larger corpora require smaller k.

b Given the n-gram language model in the table below:



	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

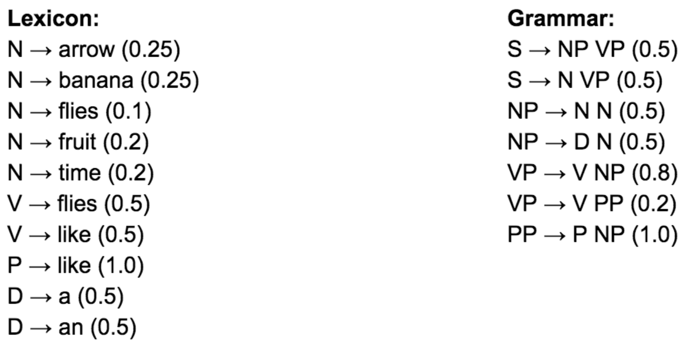
Assume the following additional probabilities of words starting sentences:
`P(i|<s>)=0.5, P(eat|<s>)=0.3, P(spend|<s>)=0.2.`

i) In each of the cases below, use this language model to generate the next word(s) to follow the prompts given, where each `...` indicates one word to be generated. Show the calculations you used in your answer:
1) `i ... ... ...`
2) `spend ... ...`
3) `i eat chinese ...`
4) `english ...`

ii) Using the same training corpus and given the prompt `spend lunch`, how could you change the language model to avoid generating gibberish sequences such as `spend lunch i`?

iii) Starting with the same prompt (e.g. the word '`i`'), suggest a strategy that would allow generating different follow up sequences using the language model in the table above.

c Given the following probabilistic grammar and lexicon:



Lexicon:

N → arrow (0.25)
N → banana (0.25)
N → flies (0.1)
N → fruit (0.2)
N → time (0.2)
V → flies (0.5)
V → like (0.5)
P → like (1.0)
D → a (0.5)
D → an (0.5)

Grammar:

S → NP VP (0.5)
S → N VP (0.5)
NP → N N (0.5)
NP → D N (0.5)
VP → V NP (0.8)
VP → V PP (0.2)
PP → P NP (1.0)

i) Provide the CKY parse matrix for the following sentence `fruit flies like a banana`

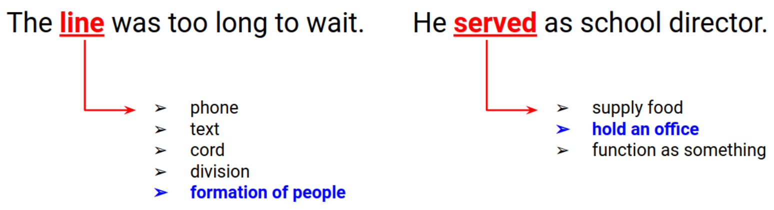
ii) Draw all the parse trees of the sentence `time flies like an arrow` using the grammar above and indicate the most probable parse tree.

d Let $T(l)$ be the number of steps (or the time taken) by the CKY algorithm to parse a sentence of length l. Complete the following table.

l	1	2	3	4	5	6
$T(l)$	1	3	7	14	**?**	**?**

The four parts carry, respectively, 15%, 45%, 30%, and 10% of the marks.

2 a In the problem of word sense disambiguation, given a sentence and **one** ambiguous word which is indicated as such, the task is to pick one among the possible senses (meanings) of the word, for example:



Consider the following settings: In a corpus of 100,000 sentences, each sentence is annotated with a single ambiguous word and its sense. In total there are 100 distinct ambiguous words, and each of them can have 2-10 different meanings. Some of these ambiguous words are more frequent than others, with the rarest one appearing in only 30 sentences, and the most frequent one in 5,000 sentences. Some senses are very rare, whereas others are very frequent. For example, out of 100 sentences with `line` as the ambiguous word, 90 have `cord` as the sense.

i) Assume you are using a Recurrent Neural Network (RNN) to address this problem. Would this be a many-to-one or many-to-many RNN? Explain why.

ii) What will the possible outputs (labels) be and which loss function would you use to train the network?

iii) Is the accuracy over all words and senses – i.e. sum of correct predictions divided by number of samples in the test set – a good way to measure the performance of the model? Explain why.

b Consider the task of hate speech detection in social media posts for 100 languages, from high-resource, such as English, to low-resource, such as Sinhala. Given a post, your multilingual model has to caterogise it as `hateful` or `not-hateful`. Assume you have labelled training data for all languages, but in different sizes: from thousands of instances for English to only a few dozen or a few hundreds for many low-resource languages. Many of these languages are not in any pre-trained representation (like BERT), so pre-trained embeddings cannot be used.

i) Of the pre-processing techniques below, which ones would you apply (indicate with ’yes’) and which you would not (indicate with ’no’)? Explain why in each case.

Technique	**Yes/No**	**Why**
Lemmatisation		
BPE		
Punctuation removal		
Case normalisation		

ii) Would you use a shared vocabulary for all languages or a vocabulary per language? Why?

iii) Consider building a CNN classifier to address this problem. Would you use filter sizes larger than two? Explain why.

The two parts carry, respectively, 55% and 45% of the marks.

3 a Consider the following text as the training corpus of a language model:

$X =$

```
The Sinclair Scientific Programmable was introduced in 1975 , with the
same case as the Sinclair Oxford . It was larger than the Scientific , at
73 by 155 by 34 millimetres , and used a larger  battery , but could also
be powered by mains electricity . It had 24 @-@ step programming
abilities , which meant it was highly limited for many purposes . It also
lacked functions for the natural logarithm and exponential function .
```

At timestep t of training, we denote the probability of the next word X_t by $P(X_t|X_{<t})$. Please answer the following questions assuming that the vocabulary size is $V = 32,000$ and the dimension for embeddings is $E = 600$.

i) Construct a feed-forward language model (FFLM) that has a context window of $C = 3$ words. We approximate the context $X_{<t}$ by concatenating the previous word embeddings. Assuming that the network has only one embedding layer and one output layer, what is the total number of learnable parameters?

ii) For the above FFLM, increasing the context size C by 1 increases the number of parameters by $\delta = VE = 19.2$ million. What would you add to the model to drastically minimise the impact of increasing C? Indicate the new δ value for the modified FFLM.

b Assume that we have a GRU-based RNNLM that we train on the above corpus X. The dimensions for word embeddings and GRU states are E and H, respectively. The vocabulary size is V.

i) A GRU has a *reset gate* that is defined as follows:

$$r_t = \sigma(W_{ir}x_t + W_{hr}h_{(t-1)} + b_r)$$

Provide the size ($\mathbb{R}^{?\times?}$) of W_{ir}, W_{hr} and b_r parameters above by making sure that the matrix multiplications are well defined (h_t's and x_t's are column vectors). Why is *sigmoid* (σ) non-linearity is preferred over alternatives such as *ReLU*?

ii) Assume that we want to compute the probability of the next word being "electricity" i.e. $P(\text{"electricity"}|X_{<t})$ with the RNNLM and you just consumed the word "mains" as the input. P_t is the final probability distribution over the items in the vocabulary. Provide answers (in the form of numbers, function names, words, symbols and so on) for each of the placeholders enumerated [1] to [6].

$$\begin{aligned} x_{t-1} &= \text{EMBEDDINGS}(\text{"[1]"}) & x_{t-1} &\in \mathbb{R}^{[2]} \\ h_t &= \text{GRU}(x_{t-1}, h_{t-1}) & h_t &\in \mathbb{R}^{[3]} \\ P_t &= [4]\,(W\,h_t) & W &\in \mathbb{R}^{[5]\times[6]} \end{aligned}$$

c Suppose that you have a vanilla RNN-based encoder-decoder NMT without attention. The encoder encodes the source word embeddings into $H = \{h_1, \ldots, h_S\}$. A single vector summary $v = f(H)$ is then obtained using a function f. Finally, v is used to initialise the decoders' hidden state.

i) Which of the following choices for $f()$ is a better way to represent source information – especially from the back-propagation perspective – and why? The choices are using (i) the last state $v = f(H) = h_S$ or (ii) the average state $v = f(H) = \frac{1}{S}\sum_{i=1}^{S} h_i$.

ii) During NMT training, the model parameters are optimised to maximise the joint likelihood of source and target sentence pairs. However at test time, we measure model's performance using automatic metrics such as BLEU. Why is it that we don't optimise the network to directly maximise such translation metrics?

d Assume that you have the following batch during NMT training.

SOURCE (X)	TARGET (Y)
two houses and fire	deux maisons et du feu
a dog running on the beach	un chien courant sur la plage
a dog shakes off water	un chien s' ébroue

i) Write the matrix $M \in \mathbb{R}^{N \times K}$ that marks the occupied (corresponding to actual tokens in the sentences) positions with 1 and unused (corresponding to padded) positions with 0 for the source sentences. Provide the values for N and K, which denote the *batch size* and the *maximum sequence length*, respectively.

ii) Recall that for a single training example, the NMT loss is the sum of log-probabilities for every token prediction. How many log-probability terms would we have for the above sample batch with three training examples?

The four parts carry, respectively, 30%, 30%, 20%, and 20% of the marks.

4 a Assume you are processing a large corpus to extract word representations.

i) List three different neural models based on the autoregressive assumption that can be used to extract word representations. Briefly explain each model.

ii) List two different neural models based on autoencoding assumption that can be used to extract word representations. Briefly explain each model.

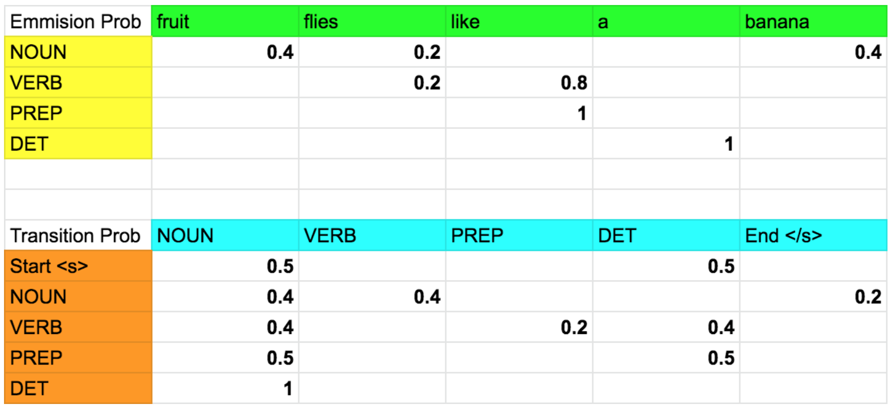
iii) Discuss one advantage and one disadvantage of models based on the autoencoding assumption compared to models based on the autoregressive assumption.

b Given a rumour detection dataset, we want to carry out neural topic modelling on both rumour texts $\mathbb{D}^1 = \{d^1_1, d^1_2, ..., d^1_{S_1}\}$ (positive cases) and non-rumour texts $\mathbb{D}^0 = \{d^0_1, d^0_2, ..., d^0_{S_0}\}$ (negative cases).

i) Write down the variational lower bound for the log-likelihood of the documents $\mathbb{D}^1$ and $\mathbb{D}^0$. Consider using $p(w|z^1)$ and $p(w|z^0)$ to represent the two distributions over words given rumour topics and non-rumour topics.

ii) Without knowing whether d^* is a rumour text, how do we make use of the topics z^1 and z^0 to infer the label (rumour or non-rumour) of d^*?

c Consider the following Hidden-Markov Model (HMM) for Part-of-Speech (PoS) tagging:



Emmision Prob	fruit	flies	like	a	banana
NOUN	0.4	0.2			0.4
VERB		0.2	0.8		
PREP			1		
DET				1	

Transition Prob	NOUN	VERB	PREP	DET	End </s>
Start <s>	0.5			0.5	
NOUN	0.4	0.4			0.2
VERB	0.4		0.2	0.4	
PREP	0.5			0.5	
DET	1				

Apply the Viterbi algorithm to PoS tag the following sentence. As part of your answer, draw the matrix indicating the probabilities for each state. Give the final tags for each word: `fruit flies like a banana .`

d Consider applying Transformers to learn a masked language model from a corpus that contains extremely long sequences. In order to capture the long term context, it is not recommended to simply cut the long sequence into independent segments. List three possible strategies to deal with this issue that could lead to more effective and/or efficient learning.

The four parts carry, respectively, 35%, 30%, 20%, and 15% of the marks.