IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2023

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

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

PAPER COMP70016=COMP97115

NATURAL LANGUAGE PROCESSING

Thursday 23rd March 2023, 10:00
Duration: 90 minutes

Answer ALL THREE questions

Paper contains 3 questions
Calculators required

1 a We have a small skip-gram model for learning word embeddings. The vocabulary contains only 3 words, with embeddings of length 2. You can find the target and context embeddings for each word in the table below. There are no other parameters (e.g. no biases) in the model.

Word	**Target embedding**		**Context embedding**	
rabbit	0.5	-0.8	-0.2	0.9
eats	-0.6	0.3	0.4	-0.1
writes	-0.4	0.2	-0.1	-0.7

You are training this model on text “my rabbit eats carrots”, “rabbit” is currently the target word and “eats” is chosen as the context word.

i) Which word does the model predict as the most likely context word for “rabbit”? Demonstrate the choice with calculations.

ii) Calculate the model loss for the given example when using softmax output and categorical cross-entropy.

iii) Calculate the loss of the model when using negative sampling, with “writes” as the only negative sample.

iv) We use each of these losses to perform 1 step of gradient descent to optimise the model. How many parameters get updated in each case?

Note: For loss calculation, use a loss equation that would be minimised for optimising the model (multiply with -1 if necessary).

b We want to use an HMM for performing the inverse of Part-of-Speech tagging: given a sequence of tags $T = t_1, \ldots, t_n$ we want to predict the words $W = w_1, \ldots, w_n$.

i) With a bigram approximation, give the equations for the transition and emission probabilities of an HMM predicting words W given tags T?

ii) If we were to use a trigam approximation to predict words W given tags T, how would the emission and transition probabilities change? Give the formula for transition and emission probabilities for this case.

iii) Consider the application of the Viterbi algorithm to the bigram approximation of the HMM model, in order to obtain the best sequence of words $\hat{W}$ given tags T. What is the asymptotic runtime of the algorithm in terms of the possible number of words $|W|$?

iv) Explain why beam search as an alternative to Viterbi can help with scaling to larger vocabulary $|W|$ when predicting words W from given tags T? (1-2 sentences)

v) Give 2 methods for dealing with previously unseen words during PoS tagging? (1 sentence each)

The two parts carry, respectively, 60% and 40% of the marks.

2 The table below contains bigram counts for our training data, showing the number of occurrences of each word in the top row when preceded by the words in the left most column.

For example, the number highlighted in bold is the number of occurrences of the word ***all*** after following the word ***love***.

	we	all	love	doing	NLP
we	0	100	0	20	0
all	50	0	120	100	0
love	300	***250***	3	150	200
doing	50	100	250	0	200
NLP	450	50	50	50	150

The next table contains unigram counts for each word:

Words:	we	all	love	doing	NLP
Unigram counts:	2,000	600	1,500	4,000	2,000

a For a bigram model, find the likelihood of the test sentence ‘we all love doing NLP’ given $P(we| < S >) = 0.1$ and $P(< /S > |NLP) = 0.2$.

b Find the perplexity of the test sentence ‘we all love doing NLP’. Include $P(< /S > |NLP)$ in your calculation.

c What would $P(all|we)$ change to if we included +K smoothing (with K=1). Assume there are 1,000 words in the vocabulary.

d State the relationship between perplexity and cross-entropy. Then demonstrate this relationship with calculations, showing how the equation for perplexity can be derived using the equation for cross-entropy. Assume cross-entropy is taken to the base *e*.

e Assume we have a unidirectional (single layered) LSTM, with 100-dimensional hidden states and 200-dimensional word embeddings. Excluding the embedding layer and any output layer, how many parameters do we have in the model? Assume there are no bias terms in the LSTM.

The five parts carry, respectively, 30%, 10%, 10%, 25%, and 25% of the marks.

3 a The BLEU score is a commonly used metric in generation systems. However, it has some drawbacks.

i) Identify 2 shortcomings of the metric and their implications (1-2 sentences for each shortcoming)

ii) What are 2 other metrics that you could use that would overcome these shortcomings? How would they help? (2-3 sentences for each metric)

b You have successfully trained a conversational agent/chat bot. However, when using the model for inference, you notice that whenever you give it the same prompt, it is always giving you the same response. Why is this happening? How might you encourage your model to give you non-deterministic outputs? Describe how this method might work. (2-4 sentences).

c When comparing Transformers and RNNs, what is the difference between how the position information for a token is represented in these models? (2-4 sentences)

d The following question tests your understanding of Transformer models. You have been given a translation dataset with 50,000 aligned sentence pairs in source and target language. An arbitrary sentence in the source language has length S, and the corresponding aligned sentence in the target language has length T. Your source language has a vocabulary size of 30,000 (V_{src}), and your target language has a vocabulary size of 20,000 (V_{trg}). You choose the hidden state dimensionality of your transformer to be d=500.

Where relevant, you may provide your answers to the dimension related questions below either algebraically, or with raw numbers.

i) What are the dimensions of the embedding matrix of this model?

ii) You will then run self-attention on your encoded tokens. Provide a very brief description about what self-attention does (1-2 sentences)

iii) What dimensions would the matrix of attention weights be in this self-attention?

iv) Once your inputs have run through your encoder stack, you will start decoding. What teacher forcing ratio, if any, is normally used when training a transformer model for machine translation?

v) At some point during decoding, we will run cross attention between the encoder outputs and the current decoder state. Cross attention is given by the formula: $z = \text{softmax}(\frac{QK^T}{\sqrt{d}})V$. For the variables Q, K and V, identify whether that variable is a function of the encoder outputs, or the current decoder state.

vi) What are the dimensions of matrix of attention weights when performing cross-attention?

The four parts carry, respectively, 30%, 15%, 15%, and 40% of the marks.