CS9180_B

THE UNIVERSITY OF WARWICK

Standard examination: Summer 2022
CS9180_B Natural Language Processing

Time allowed: 2 hours.

This exam paper contains SIX questions worth 25 marks EACH.

Answer **FOUR** questions ONLY.

Read carefully the instructions on the answer book and make sure that the particulars required are entered on **each** answer book.

Approved calculators are allowed.

---

1. (a) Write regular expressions for the following languages:
   i. the set of all alphabetic strings. [1]
   ii. the set of all vowels. [1]
   iii. the set of all lowercase alphabetic strings starting with '`a`'. [1]
   iv. all strings that start at the beginning of the line with an integer and end at the end of the line with a word. Assume that a word contains only alphabetic characters. [3]
   v. the set of all numbers. Numbers may contain decimal and thousand separators. The regular expression should match: [3]
   `30`
   `35,020.17`
   `15.2`
   but it *should not* fully match
   `24.120,17` (comma and period in wrong position)
   `12,43.5` (thousand separator in wrong place)

   (b) What are the (three) text pre-processing steps required by all NLP tasks? Give a brief description of each and the challenges each of these involve. [12]

   (c) What is the difference between lemmatisation and stemming? Give an example where the outcome of lemmatisation is different from the outcome of stemming. [4]

2. (a) What are N-gram language models? [2]

   (b) Describe one way of performing extrinsic evaluation of language models, and another way of performing intrinsic evaluation. What is a disadvantage of each of these evaluation techniques? [6]

   (c) Consider the following corpus:

   ```
   <s>I am Will </s>
   <s>Will I am </s>
   <s>I will eat anything </s>
   ```

   i. What is the probability $P(Will|am)$? [1]

   ii. What is the continuation probability of the word '`Will`'? [1]

   iii. What is the interpolated Kneser-Ney smoothing of $P(Will|am)$? Assume the fixed discount is 0.5. [2]

   iv. Write all non-zero trigram probabilities for the above corpus. [5]

   v. What do you observe for the majority of them? [1]

   vi. How can you get more realistic estimates of probabilities for these trigrams? [1]

   (d) i. Calculate the probability of the sentence '`i want to eat lunch`'. Give two probabilities, one using the bigram probabilities in Figure 1, and another using the add-one smoothed bigram probabilities in Figure 2. You can disregard the probabilities $P(i| < s >)$ and $P(< /s > |lunch)$ for the purpose of this exercise. [4]

	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

Figure 1: Bigram probabilities for eight words.

	i	want	to	eat	chinese	food	lunch	spend
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058

Figure 2: Add-one smoothed bigram probabilities for eight word.

   ii. Which of the two probabilities you computed in the previous exercise is higher, unsmoothed or smoothed? Explain why. [2]

3. (a) i. What is a one-hot word vector? [1]

ii. Give the one-hot representation of the following two sentences. Assume they are the only sentences in your corpus. No linguistic pre-processing required. [4]

1. `the world is your oyster`
2. `enthusiasm moves the world`

iii. Give two disadvantages of the one-hot representation. [2]

(b) i. What is overfitting? [1]

ii. What is the difference between microaveraged and macroaveraged evaluation? [2]

iii. Describe three ways of dealing with imbalanced data in text classification. [3]

iv. How do we compute the similarity between two word vectors using embeddings?? [2]

v. What is a continuous bag-of-words model? [1]

vi. What is a skipgram model? [1]

vii. What is the input to word2vec and what is the output? [3]

(c) What is the difference between generative and discriminative models? Is the Maximum Entropy classifier a generative or a discriminative model? Illustrate this with the algorithm's objective. [5]

4. (a) i. What is Part-of-Speech (POS) tagging and why is it considered as a sequential problem? Give examples to illustrate your answer. [4]

   ii. What is the label bias problem (use an example) and how can it be mitigated? [4]

   (b) i. Given the Penn Treebank tagset in the appendix, correct the tagging errors in the following POS tagged sentences (one error per sentence): [4]
      1. How/`WRB` many/`JJ` students/`NNS` study/`VBP` at/`IN` Warwick/`NN`
      2. The/`DT` truth/`NN` is/`VBZ` more/`RBR` important/`RB` than/`IN` the/`DT` facts/`NNS`
      3. If/`IN` you/`PRP` can/`VB` dream/`VB` it/`PRP`, you/`PRP` can/`VB` do/`VB` it/`PRP`
      4. Few/`JJ` people/`NN` arrived/`VBD` late/`RB` for/`IN` the/`DT` exam/`NN`

   ii. What is parsing? [2]

   iii. What are grammatical constituents? [2]

   iv. What are context-free grammars and what are they designed for? [2]

   v. What do context-free grammars consist of? [2]

   vi. Given the tagset and grammar rules provided in the appendix, create a CFG parse tree for the sentence below: [5]

   `I will have lunch on the flight to Barcelona`

5. (a) What is cross-validation? [2]

   (b) Describe three different ways of assessing which features are good for our classifier. [3]

   (c) How is semi-supervised classification different from supervised classification? [2]

   (d) What are two different settings to develop multiclass classifiers? [2]

   (e) What is multilabel classification and how can we achieve it? [1]

   (f) You want to train a neural network for sentiment classification and domain detection. Your training data consists of customer reviews on restaurants and hotels. The training labels include the rating scores (1 to 5 stars rating) and the domain label ('Restaurant' or 'Hotel'). Your neural network consists of an input layer with 4,096 units, a hidden layer with 2,048 units, and an output layer with 2 units (one for rating score, one for domain label). You use the ReLU activation function for the hidden units and no activation function for the outputs (or inputs).

   You train your network with the cost function:

$$J = J_{\text{sentiment}} + J_{\text{domain}}$$

   where $J_i = \frac{1}{2}(y_i - z_i)^2$, here i denotes the sentiment class or domain class. Use the following notation:

   - $\boldsymbol{x}$ is a training document (input) vector with a 1 component appended to the end, $\boldsymbol{y}$ is a training label (input) vector, and $\boldsymbol{z}$ is the output vector. All vectors are column vectors.
   - $r(\gamma) = \max\{0, \gamma\}$ is the ReLU activation function, $r'(\gamma)$ is its derivative (1 if $\gamma > 0$, 0 otherwise), and $r(\boldsymbol{v})$ is $r(\cdot)$ applied component-wise to a vector.
   - $\boldsymbol{g}$ is the vector of hidden unit values before the ReLU activation functions are applied, and $\boldsymbol{h} = r(\boldsymbol{g})$ is the vector of hidden unit values after they are applied (but we append a 1 component to the end of $\boldsymbol{h}$).
   - V is the weight matrix mapping the input layer to the hidden layer; $\boldsymbol{g} = V\boldsymbol{x}$.
   - W is the weight matrix mapping the hidden layer to the output layer; $\boldsymbol{z} = W\boldsymbol{h}$.

   i. Calculate the number of parameters (weights) in this network. Be sure to account for the bias terms. [3]

   ii. Derive $\frac{\partial J_i}{\partial W_{ij}}$, where J_i is the cost function for the sentiment class or domain class. [4]

   iii. Derive $\frac{\partial J_i}{\partial W}$, where the weight matrix $W \in \mathbb{R}^{2\times 2049}$. [4]

   iv. Derive $\frac{\partial J_i}{\partial V_{jk}}$. [4]

6. (a) Information Extraction:
   i. What is named entity recognition? [2]
   ii. What is relation extraction? [2]
   iii. Identify the named entities in the sentences below and identify their types. [3]

```
How many students study at Warwick?
The truth is more important than the facts.
If you can dream it, you can do it.
Few people arrived late for the exam.
I will have lunch on the flight to Barcelona.
```

   (b) Recommender Systems:
   i. List two advantages and two disadvantages of content-based collaborative recommender systems. [4]
   ii. Define cold start and popularity bias in recommender systems. [2]
   iii. Word2Vec represents a family of embedding algorithms that are commonly used in a variety of contexts. Suppose in a recommender system for online shopping, we have information about co-purchase records for items $x_1, x_2, \cdots, x_n$ (for example, item x_i is commonly bought together with item x_j). Explain how you would use ideas similar to Word2Vec to recommend similar items to users who have shown interest in any one of the items. [4]

   (c) Text Summarisation:
   i. Define 'extractive summarisation' and 'abstractive summarisation', and describe at least one challenge characterising each of the two approaches. [2]
   ii. List and describe the main steps involved in single- and multi-document extractive summarisation. [6]

Appendix

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coordinating conjunction	*and, but, or*	PDT	predeterminer	*all, both*	VBP	verb non-3sg present	*eat*
CD	cardinal number	*one, two*	POS	possessive ending	*'s*	VBZ	verb 3sg pres	*eats*
DT	determiner	*a, the*	PRP	personal pronoun	*I, you, he*	WDT	wh-determ.	*which, that*
EX	existential 'there'	*there*	PRP$	possess. pronoun	*your, one's*	WP	wh-pronoun	*what, who*
FW	foreign word	*mea culpa*	RB	adverb	*quickly*	WP$	wh-possess.	*whose*
IN	preposition/ subordin-conj	*of, in, by*	RBR	comparative adverb	*faster*	WRB	wh-adverb	*how, where*
JJ	adjective	*yellow*	RBS	superlatv. adverb	*fastest*	$	dollar sign	*$*
JJR	comparative adj	*bigger*	RP	particle	*up, off*	#	pound sign	*#*
JJS	superlative adj	*wildest*	SYM	symbol	*+,%, &*	"	left quote	' or "
LS	list item marker	*1, 2, One*	TO	"to"	*to*	"	right quote	' or "
MD	modal	*can, should*	UH	interjection	*ah, oops*	(	left paren	[, (, {, <
NN	sing or mass noun	*llama*	VB	verb base form	*eat*	)	right paren	],), }, >
NNS	noun, plural	*llamas*	VBD	verb past tense	*ate*	,	comma	,
NNP	proper noun, sing.	*IBM*	VBG	verb gerund	*eating*	.	sent-end punc	. ! ?
NNPS	proper noun, plu.	*Carolinas*	VBN	verb past part.	*eaten*	:	sent-mid punc	: ; ... – -

Figure A.1: The Penn Treebank tagset.

```
S -> NP VP
NP -> PRP
NP -> NNP
NP -> DT Nominal
NP -> Nominal
NP -> NP PP
Nominal -> Nominal NN
Nominal -> NN
VP -> VB
VP -> MD VP
VP -> VB NP
VP -> VP NP PP
VP -> VP PP
PP -> TO NP
```

Figure A.2: CFG rules