The University of Melbourne

Department of Computing and Information Systems

COMP90042
Web Search and Text Analysis
July 2018

Identical examination papers: None

Exam duration: Two hours

Reading time: Fifteen minutes

Length: This paper has 6 pages including this cover page.

Authorised materials: None

Calculators: Not permitted

Instructions to invigilators: Students may not remove any part of the examination paper from the examination room. Students should be supplied with the exam paper and a script book, and with additional script books on request.

Instructions to students: This exam is worth a total of 50 marks and counts for 50% of your final grade. Please answer all questions in the script book provided, starting each question on a new page. Please write your student ID in the space below and also on the front of each script book you use. When you are finished, place the exam paper inside the front cover of the script book.

Library: This paper is NOT to be held in the Baillieu Library.

Student id:

Examiner's use only:

Q1	*Q2*	*Q3*	*Q4*	*Q5*	*Q6*	*Q7*	*Q8*	*Q9*

Continued overleaf ...

COMP90042 Web Search and Text Analysis

Semester 1, 2018

Total marks: 50

Students must attempt all questions

Section A: Short Answer Questions [15 marks]

Answer each of the questions in this section as briefly as possible. Expect to answer each sub-question in no more than a line or two.

Question 1: General Concepts [8 marks]

a) Explain why the F_1 score is widely used in evaluating text processing methods, but is much less widely used in retrieval. [1 mark]

> The F-score combines precision and recall, which measure the accuracy of the predictions, and how much of the gold standard has been correctly recovered, respectively. In NLP recall tends to be easier to measure, as the space of classes is typically small, whereas in IR the set of documents is massive, and it's infeasible to have them exhaustively annotated for relevance (1 mark). Thus IR uses precision based metrics.

b) Name one task where the "bag-of-words" document representation is appropriate, and another task where "bag-of-words" is inappropriate. In each task, explain why this is the case. [2 marks]

BOW (1 mark for any; 0.5 = task, 0.5 = reason):

- Text classification (including specific instances such as sentiment analysis)
- Information retrieval
- Topic models
- . . . based on reasoning that the words carry the key information about the document, more so than their order, phrases, or other linguistic structures

Not BOW (1 mark for any; 0.5 = task, 0.5 = reason):

- Parsing
- Machine Translation
- Language Modelling
- . . . any structured prediction task, as to model structure requires knowledge of how the words relate to their context

c) Contrast dependency parsing with syntactic phrase-structure parsing. Identify at least one key similarity and one important difference. [2 marks]

Similarity (1 mark for any):

- Both based on the idea of a syntax tree
- Dynamic programming style algorithms can be used for parsing
- Dep parsing can be reduced to phrase-structure parsing using grammar transform
- Notion of "head" in constituency parsing highly similar to dependency

Differences (1 mark for any):

- Formulation based on head-child dependencies in dependency grammars, not text consituent spans
- Specific transition base parsing algorithms for dependency parsing
- Possibility of handling non-projective trees in dependency parsing, which means subtrees do not correspond to a contiguous span in the text

d) Explain why "word alignment" in machine translation is typically framed as an unsupervised learning problem. [1 mark]

Typically bilingual translation data is readily available at the sentence level, but not at the phrase or word level. Word alignment seeks to "fill in" the missing values using unsupervised learning (1 mark). The way this works is using the EM algorithm, based on a model which connects word based translation actions to a sentence generation probability model.

e) Unsupervised HMMs can be trained using Expectation-Maximisation. Explain the difference between hard EM and soft EM. Why is soft EM considered better? [2 marks]

Differences (1 mark for any):

- Viterbi vs. Forward-Backward
- Argmax tags vs. full posterior over tags
- True counts vs. expected counts
- ...

Why soft EM is better (1 mark): because it estimates full distributions for each tag instead of just taking the most probable one. This gives more information to the parameter estimation procedure, especially for highly ambiguous tags.

Question 2: Information Retrieval [4 marks]

a) Name a situation where "compression" is important in information retrieval, and for your chosen situation, explain what property of the data allows for compression. [1 mark]

Inverted index postings list compression (0.5 marks) dramatically reducing the size of the search index. We utilise the property that many of the numbers (gaps or frequencies) in the postings list are small (0.5 marks). Thus, a compression codec such as vbyte which spends less bits encoding small numbers helps reduce space usage of the index structure.

b) What is pseudo relevance feedback and why is it useful? [1 mark]

Pseudo relevance feedback uses the top-K results of an initial query to find additional important/informative query terms contained in many of these documents (0.5 marks). This is useful to help with vocabulary mismatch problems (toxin vs poison, danger vs hazard) (0.5 marks).

c) The "mean average precision" metric is often used for evaluation of retrieval effectiveness. State its formulation, and explain why it is more appropriate than precision at rank k. [2 marks]

Mean average precision (MAP) is the average precision at each point in the ranking where a relevant document occurs: $MAP(\vec{f}, k, q) = \frac{1}{R_q} \sum_{i=1}^{i=k} f_i p@i(\vec{f})$, where $\vec{f}$ is the relevance vector of the documents in the ranking (relevance is denoted with a binary value), k is the number of retrieved documents, q is the set of relevant documents and R_q is the number of relevant documents. The precision term $f_i p@i(\vec{f})$ at point i is the average number of relevant documents in the top i elements, $f_i p@i(\vec{f}) = \frac{1}{i} \sum_{j=1}^{j=i} f_i$. (1 mark for the equation above, or equivalent.)

MAP is more appropriate thatn precision at k as it also incorporates the position in the ranking, such that when e.g., of 10 returned documents with 5 valid, MAP will reward systems that put the 5 valid documents high in the ranking with a higher score. (1 point)

Question 3: Distributional Semantics [3 marks]

a) Explain two important benefits of using vector representations for words, instead of simple discrete features. [2 marks]

1 mark for each (2 max):

- accounts for similarities between words
- can use unlabelled corpora to add information in supervised models
- low-dimensional vectors can be faster to use compared to high-dimensional ones

b) The "Word2Vec skip-gram" algorithm and "latent semantic analysis" both learn vector representations of words. Explain a key difference between these two approaches. [1 mark]

1 mark for any:

- LSA is based on counts and matrix factorisation, skip-gram is based on predicting the context words
- LSA usually preprocess matrices using PMI, skip-gram doesn't

Section B: Method Questions [17 marks]

In this section you are asked to demonstrate your conceptual understanding of the methods that we have studied in this subject.

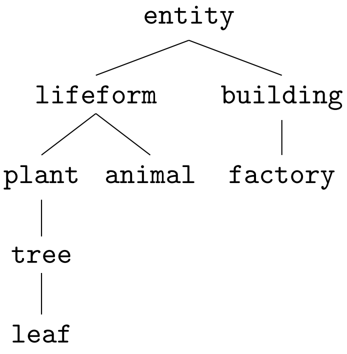
Question 4: Lexical semantics [5 marks]

a) Fill in this sentence with the appropriate *-nym*: *leaf* is a _______________ of *plant* (in the sense of vegetation). [1 mark]

Meronym

b) Given the hierarchy below calculate the following similarity metrics: [2 marks]

- Path similarity:
  - `plant` and `factory`
  - `leaf` and `animal`
- Wu-Palmer similarity:
  - `plant` and `factory`
  - `leaf` and `animal`



Path similarities (1 mark for both):

- Path(plant, factory) = 1/5
- Path(leaf, animal) = 1/4

Wu-Palmer similarities (1 mark for both):

- Path(plant, factory) = 2*1 / 3+3 = 2/6 = 1/3
- Path(leaf, animal) = 2*2 / 5+3 = 4/8 = 1/2

c) How is "Lin similarity" calculated? Mention all resources required and the major steps in the calculation. [2 marks]

Lin similarity uses frequency statistics (0.5). By leveraging a corpus, it add frequency information as an Information Content (IC) component (0.5). IC is calculated by summing frequencies for each word in a concept (0.5). The full measure calculates IC for lowest common subsumer and then divide it by the product of the ICs of the two concepts we are getting the similarity. (0.5)

Question 5: Grammars and Parsing [7 marks]

This question is based on the "CYK" algorithm for PCFGs, shown below in pseudo-code:

```
function parse-CYK(w, G):
1       for j in 1 .. |w|
2           for all A -> wj in grammar
3               set chart[j-1,j,A] = P(A -> wj)
4           for i in j-1 .. 0 (descending)
5               for k in i+1 .. j-1
6                   for all A -> B C in grammar
7                       prob = P(A -> B C) * chart[i,k,B] * chart[k,j,C]
8                       if prob > chart[i,j,A] then
9                           chart[i,j,A] = prob
10                          back[i,j,a] = (k, B, C)
11      return build-tree(back, |w|, S)
```

a) The "CYK algorithm" assumes the grammar is in "Chomsky Normal Form (CNF)". What is CNF? Explain with relation to the above code, why this restriction is required. [2 marks]

> CNF means that all productions are of the form $A \rightarrow BC$ or $A \rightarrow a$ where capitals denote non-terminals, and lowercase denotes terminals (1). This restriction is needed so that line 3 can handle the unary rules, without needing recursion (0.5); and lines 4-10 can handle binary productions, where both child symbols must cover adjacent spans (0.5). Ternary or non-adjacent non-terminals would require a more complex algorithm for parsing, with higher computational complexity.

b) What is stored in the `chart`? [1 mark]

> The chart holds the best partial parse for a span of the sentence, based on the root of the parse having a given non-terminal. (0.5 = best parse; 0.5 = link to indexing of chart)

c) Explain how the iteration order in the for-loops above is critical for the correct updating of the chart. [2 marks]

> This encodes a bottom up search (1 mark), such that all smaller spans are processed before larger spans (1 mark).
>
> To give more details: the iteration order considers first the right end-point of the span (j; line 1) which goes from left to right over the sentence, then the start point which marches left (i; line 4). Finally the midpoint moves between these (k; line 5). In such a way all sub-spans needed for line 7 have been computed by the time the values are needed inside a larger span.

d) Provide code for the algorithm `build-tree`, which recovers the best scoring tree, as invoked on line 11. Feel free to use Python or pseudo-code syntax. [2 marks]

```
function BUILD-TREE(back, i=0, j=$|w|$, A=S)
    if i+1 == j then
        return Leaf(A)
    else
        k, B, C = back[i,j,A]
        left = build-tree(back, i, k, B)
        right = build-tree(back, k, j, C)
        return Node(A, left, right)
        end if
    end function
```

1 mark = on the right track, but incorrect 2 marks = working solution (at level of pseudo-code)

Question 6: Information Extraction [5 marks]

Consider the following toy data, composed of only one sentence, its corresponding Named Entity annotation and a gold set of relations extracted from it:

- `Amanda Palmer is a singer-songwriter born in 1976 in New York City, New York, US.`
- `[Amanda Palmer]`$_{\textbf{PER}}$ `is a singer-songwriter born in [1976]`$_{\textbf{TIME}}$ `in [New York City]`$_{\textbf{LOC}}$`, [New York]`$_{\textbf{LOC}}$`, [US]`$_{\textbf{LOC}}$`.`
- Gold relations:
  - `year-of-birth(Amanda Palmer, 1976)`
  - `place-of-birth(Amanda Palmer, New York City)`
  - `city-state(New York City, New York)`
  - `state-country(New York, United States of America)`

a) Suppose you want to train a Named Entity Recogniser using an HMM. Rewrite the NE annotated sentence into a sequence of (word, tag) elements using one of the schemes you learned in class. Write your answer in the following format: `word1/tag1 word2/tag2 ...` [2 marks]

> This uses IOB. A similar answer using IO is also acceptable. `Amanda/B-PER Palmer/I-PER is/O a/O singer-songwriter/O born/O in/O 1976/B-TIME in/O New/B-LOC York/I-LOC City/I-LOC ,/O New/B-LOC York/I-LOC ,/O US/B-LOC ./O`
> 2 marks = fully correct; 1 mark = one or two mistakes; not using IO/IOB etc = 0 marks

b) The first step in Relation Extraction is to build a binary classifier that recognises if two entities have a relation or not. Assuming the example above is the only data you have available, how many positive and how many negative instances you would have in your training set for this classifier? [1 mark]

> 4 positive instances and 6 negative instances

c) The second step in Relation Extraction is to build a multi-class classifier that, given a positive entity pair, predicts the relation between them. However, even if you have a perfect classifier the relations extracted from the sentence will not match the gold relations given above. Why is this the case? How would you solve this problem, so the relations match the gold standard? [2 marks]

> The problem is the mismatch between `US` and `United States of America` (1). This can be solved by text normalisation, using a gazeteer for instance (1).

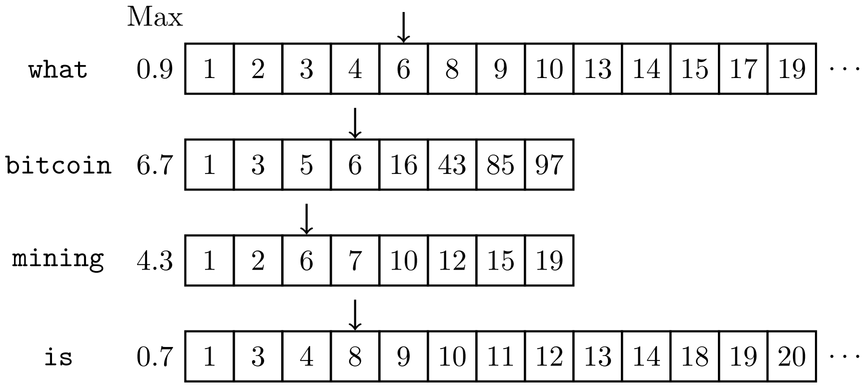
Continued overleaf ...

Section C: Algorithmic Questions [10 marks]

In this section you are asked to demonstrate your understanding of the methods that we have studied in this subject, in being able to perform algorithmic calculations.

Question 7: Document and Term ranking [4 marks]

Consider the following snapshot of an instance of the WAND top-K query processing algorithm for the query `'What is bitcoin mining'` and $K = 2$. Suppose the algorithm has just evaluated document 6 with score 4.4. Answer the following questions:


Max
what 0.9 | 1 | 2 | 3 | 4 | 6 | 8 | 9 | 10 | 13 | 14 | 15 | 17 | 19 | ···
bitcoin 6.7 | 1 | 3 | 5 | 6 | 16 | 43 | 85 | 97
mining 4.3 | 1 | 2 | 6 | 7 | 10 | 12 | 15 | 19
is 0.7 | 1 | 3 | 4 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 18 | 19 | 20 | ···

#	Score	DocId
1	8.7	1
2	4.1	3

a) What is the idea that allows the WAND algorithm to skip evaluating documents? [1 mark]

b) Was document 4 ever in the top-K score list? Explain your reasoning. [1 mark]

c) What is the next document that will be evaluated? Explain your reasoning. [2 marks]

a) We store maximum contributions which overestimate the contribution each of the query terms can have towards the score of a query (0.5). If even with this overestimated score, a document could never enter the top-K result list, we don't have to score it (0.5).

b) document 4 was never evaluated as the maximum score it could have is 0.9+0.7 (it only occurs in terms "what" and "the"). However, as we are processing Documents-at-a-Time, at this point of the query processing, documents 1 and 3 must have been evaluated already (either reason = 1 mark). Thus, it was never evaluated and it could never enter the top-K heap.

c) Long version: Document 6 is evaluated with score 4.4 and replaces document 3 in the heap as the lowest scoring document. All the pointers are forwarded such that the next smallest document is document 7 which only occurs in the term "mining". As the maximum contribution of "mining" is 4.3 (which is lower than the current lowest score 4.4) we do not evaluate 7. The next document we consider evaluating now is 8. We forward list "mining" with GEQ(8) to 8, but as it does not contain 8 it now points to 10. The sort order currently is $< what, 8 >$,$< is, 8 >$,$< mining, 10 >$,$< bitcoin, 16 >$. Thus, 10 could be the next evaluated document and we perform GEQ(10) on lists "what" and "is" so the new list order looks like this: $< what, 10 >$,$< is, 10 >$,$< mining, 10 >$,$< bitcoin, 16 >$. Thus, we evaluate document 10 next.

Short version: Instead of doing all the resorting business, we just look for the next larger number that is contained in lists such that the sum of the maximum contributions of those lists is larger than 4.4. Thus, we pick 10.

1 mark = correct answer. 1 mark = valid reasoning (even for incorrect answer)

Question 8: N-gram language modelling [6 marks]

This question asks you to calculate the probabilities for n-gram language models. You should leave your answers as fractions. Consider the following corpus with 3 "sentences":

bbaccab
aaacbba
bacabba

a) Calculate a unigram language model for this corpus. [1 mark]

$$P(a) = 9/21$$
$$P(b) = 8/21$$
$$P(c) = 4/21$$

b) Next, calculate an MLE bigram language model. Add additional symbols as needed. Under this model, what is the probability of the sentence *abc*? [3 marks]

$$\begin{array}{rrr} P(a|<s>) = 1/3 & P(b|<s>) = 2/3 & P(c|<s>) = 0 \\ P(a|a) = 2/7 & P(b|a) = 2/7 & P(c|a) = 3/7 \\ P(a|b) = 4/7 & P(b|b) = 3/7 & P(c|b) = 0 \\ P(a|c) = 2/4 & P(b|c) = 1/4 & P(c|c) = 1/4 \end{array}$$

$P(abc) = P(a| < s >) * P(b|a) * P(c|b) = 1/3 * 2/7 * 0 = 0$

Marks: -1 includes ¡s¿ sentinel
-1 one or two mistakes
-1 three or more mistakes

c) Calculate smoothed bigram probabilities for all those terms where b is the context (that is, $p(\cdot|b)$) by applying add-one smoothing. Calculate the probability of the sentence *abc* under this new model. [1 mark]

$$P(a|b) = 5/10$$
$$P(b|b) = 4/10$$
$$P(c|b) = 1/10$$

$P(abc) = P(a| < s >) * P(b|a) * P(c|b) = 1/3 * 2/7 * 1/10 = 2/210$ (Just reporting the fractions is acceptable. If reporting final probability, minor calculation mistakes are also acceptable)

0.5 for component probs; 0.5 for *abc* prob

d) Calculate smoothed bigram probabilities for terms where b is the context (that is, $p(\cdot|b)$) by interpolating it with unigram counts, using 0.5 as the interpolation weight. Calculate the probability of the sentence *abc* under this new model. [1 mark]

$$\begin{aligned} P(a|b) &= (0.5 * 4/7) + (0.5 * 1/3) = 2/7 * 1/6 = 2/42 \\ P(b|b) &= (0.5 * 3/7) + (0.5 * 2/3) = 3/14 * 1/3 = 1/14 \\ P(c|b) &= (0.5 * 0) + (0.5 * 4/21) = 2/21 \end{aligned}$$

$P(abc) = P(a| < s >) * P(b|a) * P(c|b) = 1/3 * 2/7 * 2/21 = 4/441$ (Just reporting the fractions is acceptable. If reporting final probability, minor calculation mistakes are also acceptable)

0.5 for component probs; 0.5 for *abc* prob

Section D: Essay Question [8 marks]

Question 9: Essay [8 marks]

Discuss *one* of the following options (about 1 page). Marks will be given for correctness, completeness and clarity.

- **Word sense ambiguity.** Define the problem of word sense ambiguity, with the aid of examples. Motivate why this is an important problem and a hard one to solve, and outline methods for word sense disambiguation.

  Marks are assigned based on the presence of the following components:

  1. Clarity (2 marks)
  2. Problems (2 marks) (e.g. polysemy)
  3. Examples (1 mark)
  4. Hardness (1 mark) (e.g. resources)
  5. Methods (2 marks)
     (a) Lesk
     (b) Yarowsky bootstrap
     (c) Classifiers
     (d) Predominant sense

- **Word vector learning.** Define the problem of using discrete word representations, with the aid of examples. Motivate why this is an important problem, outline methods used to solve it, how it is evaluated and how word vectors learning, and word vectors themselves, are related to other tasks in language processing.

  Marks are assigned based on the presence of the following components:

  1. Clarity (2 marks)
  2. Outline and motivation (2 marks)
  3. Methods (2 marks; any 2)
     (a) LSA/PMI etc
     (b) Word2vec
     (c) Part of larger RNN or seq2seq
  4. Evaluation by cosine over word-pair collections, analogies (1 mark; either is ok)
  5. Analysis, down-stream usage in other NLP tasks/models (1 mark)

- **Machine Translation.** A long running challenge in language processing has been the automatic translation between different languages. Discuss the key difficulties of translation, outline the sub problems and how they can be solved to create an automatic translation system, how they are evaluated, and discuss the strengths and weaknesses of these solutions.

  Marks are assigned based on the presence of the following components:

  1. Clarity (2 marks)
  2. Outline and difficulties (2 marks)
  3. Sub problems (3 marks)
     (a) Sentence alignment (1 mark)
     (b) Word alignment (1 mark)
     (c) End to end seq2seq for machine translation (1 mark)
  4. Analysis (1 mark)

(a) Morphology

(b) Word based vs contextual

— End of Exam —