The University of Melbourne

Department of Computing and Information Systems

COMP90042
Web Search and Text Analysis
June 2015

Identical examination papers: None

Exam duration: Two hours

Reading time: Fifteen minutes

Length: This paper has 12 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 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
Final Exam

Semester 1, 2015

Total marks: 50

Students must attempt all questions

Section A: Short Answer Questions [14 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 [6 marks]

1. The frequency of words in natural language is often described as having a "Zipfian distribution". Describe what this means, and outline an important consequence for text processing. [2 marks]

   Zipf's 'law' states that there is a relation between r, the rank of a word in the lexicon, sorted by frequency, and the word's frequency, f, namely $f \propto \frac{1}{r}$ (1 mark; n.b., must define f and r). This means that sentences are mostly comprised of the really common words (the, say, did, etc, with low r), but there's a long tail of many, many, uncommon words. A consequence for information retrieval is that the the long tail of uncommon words are more informative, which means we need to focus our effort on these terms (e.g., using IDF, also efficiency concerns with data structures and algorithms). (1 mark) Alternative rationale: A similar rationale applies to text classification, topic modelling and dictionary learning, among other tasks, where these uncommon words are often more important than common words. *Note we didn't cover the Zipfian distribution in the same depth this year.*

2. Provide a definition of the "mean average precision" evaluation measure, and justify why it is appropriate for evaluating ranked retrieval. [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; could state these together in one equation)
   MAP is appropriate for IR because it captures the quality of the ranking given a fixed number of returned relevant documents (contrary to precision, accuracy and F-score). For example, out of 10 returned documents only 5 could be valid. MAP will reward systems that put the 5 valid documents high in the ranking with a higher score. (1 point)

3. "Smoothing" is often important when dealing with models of text; outline two contexts where "smoothing" is used and describe its effect. [2 marks]

   Smoothing in the context of text modeling is the concept of modifying text statistics (e.g. estimates of word probabilities) in such a way that they are made more sensible from the application point of view. One common reason is to avoid zero probabilities or more generally not rely too heavily on infrequent events (1 mark). Smoothing is used

 Continued overleaf ...

for example in language modeling, where without smoothing occurrence of a word (or combination of words) that has not been observed in the training set leads to probability zero of the whole sentence. In machine translation, if a pair of words from two languages has never been observed together in a pair of aligned sentences, the word translation probability would be estimated to be zero between this pair of words. This might be an overly pessimistic strategy and so smoothing can be employed to mitigate this. (1 mark)
Other examples are acceptable.

Question 2: Information Retrieval [4 marks]

1. What are "stop-words" and why are they often discarded in information retrieval? [2 marks]

   Stop-words are typically very common words in the corpus (1/2 mark). They are often discarded because they are uninformative, and would be given a near-zero IDF weight due to their occuring in almost every document. This means they have negligible effect on cosine similarity values under a TF*IDF model (1 mark). Removing stop-words is for efficiency purposes: it saves space in the index (no need for the biggest posting lists), as well as gain speedup in algorithms' runtime (no need to process these posting lists) (1/2 mark).

2. Explain the reason why "posting lists" are typically stored in sorted order by document identifier. [1 mark]

   Posting lists are stored in a sorted order because this allows for:
   (a) efficient query processing by simultaneous traversal of posting lists corresponding to terms in a query
   (b) compressing the lists by storing the gaps between the consecutive document ids rather than the documents ids themselves
   (c) speeding up the lookup via skip lists (*not covered this year*)

   Any one of the above gives 1 mark.

3. Compression is often considered in the context of efficient on-disk storage. Describe why compression is also important for in-memory storage, in the context of information retrieval. [1 mark]

   Compression is important for in-memory storage because it is faster to process data in memory than on disk. We typically want to minimise the number of disk accesses as they are usually orders of magnitude slower than memory operations. Therefore, we can often afford the computation in decompression in return for being able to store more data in-memory.

Question 3: Text Analysis [4 marks]

1. Natural language is ambiguous in many ways; provide two examples illustrating different kinds of ambiguity in English. [2 marks]

   (a) a word can have multiple meanings, e.g. a prune can be interpreted as a dried plum or as an operation of truncating a tree
   (b) in syntactic parsing of a sentence with a grammar; e.g. a sentence "I shot an elephant in my pajamas" can be decomposed in at least two different ways, where the prepositional phrase "in my pajamas" can be either assigned to the speaker or to the elephant

Continued overleaf . . .

(c) in named entity recognition/linking, where entity occurrences are linked with the actual objects (e.g. does Paris in a sentence refer to PERSON Paris Hilton or to LOCATION the capital of France)

(d) the same meaning can be expressed in different ways using natural language ("I saw a dog," "A dog has been seen by me")

Any two of the above give 2 marks. Note this is not a comprehensive list of examples.

2. What is a "homonym" and why might they prove problematic for language processing? [1 mark]

   A homonym is a word with multiple meanings. (1/2 mark) This linguistic phenomenon poses significant challenges in many Natural Language Processing tasks, for example different meaning of one word can map to multiple different words in a different language, as encountered in machine translation. (1/2 mark) Some other examples where this is a problem are Word Sense Disambiguation, POS-tagging and Information Retrieval.

3. Outline with the aid of an example how the "IOB" method allows for sequence classification to be applied to multi-word labelling tasks such as named entity recognition. [1 mark]

   IOB is the labeling scheme of words in a sentence, where B tag denotes the beginning of the named entity, I denotes a word inside the named entity and O denotes a word outside of the named entity. In a sentence "I saw Bill Clinton" the tagging will be O O B I, thus marking Bill Clinton as a multi-word named entity.

Continued overleaf . . .

Section B: Method Questions [18 marks]

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

Question 4: Document Search [7 marks]

1. Present and contrast the algorithms for the "bi-word" and "positional" index methods for document retrieval with phrasal queries. For both methods show the algorithms for querying and index construction, and illustrate their operation with a simple example. [4 marks]

   Not covered in 2016.

2. Outline a method for compressing a positional index, and describe what property of language is being exploited by this technique. [3 marks]

   Not covered in 2016.

Question 5: Web as a Graph [4 marks]

The "page rank" and "hubs and authorities" algorithms are means of deriving document importance measures automatically from a hyper-linked corpus.

1. Explain with the aid of an example the terms "hubs" and "authorities". [1 mark]

   Not covered in 2016.

2. The "page rank" method is framed as the frequency with which a random surfer visits each web page in a collection. Provide a similar analogy for the "hubs and authorities" method, and show how this gives rise to a probabilistic model with iterative update equations,

$$h_i \leftarrow \sum_{i \to j} a_j$$
$$a_i \leftarrow \sum_{j \to i} h_j$$

   where i and j index the pages, $i \to j$ and $j \to i$ denote directed edges in the graph (hyperlinks) and $\mathbf{h}$ and $\mathbf{a}$ are vectors of hub and authority scores. [3 marks]

   Not covered in 2016.

Question 6: Markov Models [7 marks]

1. Describe the assumptions that underlie Markov models, and provide a part-of-speech tagging example showing where these assumptions are inappropriate. [2 marks]

   In a Markov model, the joint probability over a sequence of random variables is decomposed in such a way that the probability of each consecutive random variable is conditioned only on a fixed number of preceding random variables. (1 mark) This assumption is inappropriate for example in case of WH questions, where the starting WH word (e.g. why or where) makes for a useful information in determining that the last word is likely to be a verb, and that it is likely to have a final question mark. In normal sentences these are unlikely events. For sentences that are longer than the order of the Markov model this information is missing. (1 mark)

 Continued overleaf . . .

2. What classes of formal languages can be described by Markov models over word sequences? Relate this to context free grammars used in parsing. [2 marks]

   Markov models can describe regular languages, which can be described by finite state automata and regular expressions. (1 mark) Context free grammars are more powerful in that they can describe context free languages, an upper set of regular languages (all regular languages are context-free, but there are context-free languages which are not regular). (1 mark)

3. The Viterbi algorithm for hidden Markov models uses dynamic programming to compute the maximum probability path of hidden states that generates a given sequence of observations,

$$\mathbf{t}^* = \text{argmax}_{t_1,t_2,\ldots,t_{N-1},t_N}\, p(w_1, t_1, w_2, t_2, \ldots, w_{N-1}, t_{N-1}, w_N, t_N)\,.$$

   Consider the related *forward* problem of marginalising (summing) the probability over all paths for an observation sequence in order to compute the probability of the observations, i.e.,

$$p(w_1, w_2, \ldots, w_{N-1}, w_N) = \sum_{t_1,t_2,\ldots,t_{N-1},t_N} p(w_1, t_1, w_2, t_2, \ldots, w_{N-1}, t_{N-1}, w_N, t_N)\,. \qquad (1)$$

   Using a similar approach to the Viterbi algorithm, show how Equation (1) can also be solved using dynamic programming. This involves reducing Equation (1) to a recursive formulation. [3 marks]

   Let us define

$$\alpha(k, t_k) = \sum_{t_1,t_2,\ldots,t_{k-1}} p(w_1, t_1, w_2, t_2, \ldots, w_{k-1}, t_{k-1}, w_k, t_k)\,.$$

   $\alpha(k, t_k)$ can be represented as a function of $\alpha(k-1, t_{k-1})$ values:

$$\begin{aligned}
&\alpha(k, t_k) \\
&\quad = \sum_{t_1,t_2,\ldots,t_{k-1}} p(w_1, t_1, w_2, t_2, \ldots, w_{k-1}, t_{k-1}, w_k, t_k) \\
&\quad = \sum_{t_{k-1}} \sum_{t_1,t_2,\ldots,t_{k-2}} p(w_1, t_1, w_2, t_2, \ldots, w_{k-1}, t_{k-1}) p(t_k|t_{k-1}) p(w_k|t_k) \\
&\quad = (\sum_{t_{k-1}} \alpha(k-1, t_{k-1}) p(t_k|t_{k-1})) p(w_k|t_k).
\end{aligned}$$

   Which together with the base case:

$$\alpha(1, t_1) = p(t_1) p(w_1|t_1)$$

   Allows us to recursively find all α values and obtain $p(w_1, w_2, \ldots, w_{N-1}, w_N) = \sum_{t_N} \alpha(N, t_N)$. (3 marks)

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 [5 marks]

Consider the following "term-document matrix", where each cell shows the frequency of a given term in a document:

DocId	stock	share	corporate	corruption	london	barclays
doc_1	2	2	1	0	0	0
doc_2	1	3	2	0	1	1
doc_3	0	0	1	1	1	1
doc_4	0	0	0	3	4	0

This question is about using the term-document matrix for querying and to find similar terms.

1. Calculate the "document ranking" for the query *corporate corruption* using a "cosine similarity" measure over raw term frequencies. [1.5 marks]

   The query can be represented as $q = [0, 0, 1, 1, 0, 0]$. We'll now compute the cosine between this and each document:

   $$\cos(q, doc_1) = \frac{1+0}{\sqrt{2^2+2^2+1^2}} = \frac{1}{3}$$
   $$\cos(q, doc_2) = \frac{2+0}{\sqrt{3^2+2^2+1^2+1^2+1^2}} = \frac{1}{2}$$
   $$\cos(q, doc_3) = \frac{1+1}{\sqrt{1^2+1^2+1^2+1^2}} = 1$$
   $$\cos(q, doc_4) = \frac{0+3}{\sqrt{4^2+3^2}} = \frac{3}{5}$$

   Note that for ranking, you don't need to normalize by the length of the query vector, since it is the same for all documents.

   This gives a ranking of: doc_3, doc_4, doc_2, doc_1

2. Calculate the "retreival status value" for doc_2 with the query *corporate corruption* according to a smoothing unigram language model, $P(d|q) \propto \prod_{t \in q} \lambda P(t|M_d) + (1-\lambda)P(t|M_c)$. Use maximum likelihood estimates for the document specific language models M_d and corpus language model M_c and let $\lambda = 0.5$. You do not need to simplify numerical values. [2 marks]

   For $d = doc_2$, the unigram estimates are:

   $$M_d(\text{corporate}) = 2/(3+2+1+1+1) = \frac{1}{4}$$
   $$M_d(\text{corruption}) = 0$$

   The total counts over the corpus are $N = 2+2+1+3+2+1+1+1+1+1+1+1+4+3 = 24$.

   The corpus unigram estimates are:

   $$M_c(\text{corporate}) = (1+2+1)/24 = \frac{1}{6}$$
   $$M_c(\text{corruption}) = (1+3)/24 = \frac{1}{6}$$

Continued overleaf ...

Plugging into the formula, we get:

$$P(d|q) = \big(1/2 \times 1/4 + 1/2 \times 1/6\big) \times \big(1/2 \times 0 + 1/2 \times 1/6\big) = 5/24 \times 1/12 = \frac{5}{288}$$

3. Calculate the pairwise similarity between term *corporation* and all other terms (*stock*, *share*, ..., *barclays*) based on "cosine similarity". You do not need to simplify numerical values. [1.5 marks]

Since *corporation* is not listed, you might argue that the cosine similarity is undefined. However, this is a bit of a cop-out. Assuming the intention is *corporate* and there's a typo (the terms would be conflated by stemming, anyway).

First compute the term lengths

$$\begin{aligned}
||\text{corporate}|| &= \sqrt{1^2 + 2^2 + 1^2} = \sqrt{6} \\
||\text{stock}|| &= \sqrt{1^2 + 2^2 + 1^2} = \sqrt{6} \\
||\text{share}|| &= \sqrt{2^2 + 2^3} = \sqrt{13} \\
||\text{corruption}|| &= \sqrt{1^2 + 3^2} = \sqrt{10} \\
||\text{london}|| &= \sqrt{1^2 + 1^2 + 4^2} = \sqrt{18} \\
||\text{barclays}|| &= \sqrt{1^2 + 1^2} = \sqrt{2}
\end{aligned}$$

Now we can compute cosine values

$$\begin{aligned}
\cos(\text{corporate}, \text{stock}) &= (2+2)/(\sqrt{6} * \sqrt{6}) = \frac{1}{3} \\
\cos(\text{corporate}, \text{share}) &= (2+3*2)/(\sqrt{6} * \sqrt{13}) = \frac{8}{\sqrt{78}} \\
\cos(\text{corporate}, \text{corruption}) &= 1/(\sqrt{6} * \sqrt{10}) = \frac{1}{\sqrt{60}} \\
\cos(\text{corporate}, \text{london}) &= (2+1)/(\sqrt{6} * \sqrt{18}) = \frac{3}{\sqrt{108}} \\
\cos(\text{corporate}, \text{barclays}) &= (2+1)/(\sqrt{6} * \sqrt{2}) = \frac{3}{\sqrt{12}}
\end{aligned}$$

Note that with a calculator or computer we could find the ranking easily enough.

Question 8: Grammars and Parsing [5 marks]

Describe an algorithm for chart parsing with a context free grammar. Illustrate with a worked example, e.g., using a simple grammar and a short 4-7 word sentence.

Earley parsing is a top-down chart parsing algorithm which relies on dynamic programming to avoid redundant parsing.

Given a context free grammmar with a set of productions, parsing a sentence of length n means to fill a chart with n + 1 cells with edges, where each edge corresponds to a production with a dot on the left hand side, indicating how much of the rule has been consumed, and a set of indicies which indicates the beginning and end of the span that has been consumed. For example,

S $\rightarrow$ NP $\circ$ VP [0,2]

represents a situation where the NP part of the rule S $\rightarrow$ NP VP has been satisfied by the first two words of the sentence. The algorithm proceeds via three operations: the predictor, which

 Continued overleaf ...

predicts blank rules, the scanner, which reads in input, and the completer, which creates new edges where dots have been advanced. The algorithm begins by predicting, in the first column of the chart, all S rules, and any rules for nonterminals appearing first on the right hand side of S rules, and any rules for the nonterminals on the RHS of those rules, and so on, until no more rules can be predicted. Then, it advances through the sentence. At each stage:

The scanner reads in the current token by adding completed productions with the relevant nonterminal to the chart, such as

DT $\rightarrow$ *the* $\circ$ [0,1]

The completer then looks at the previous column of the chart, and looks for any other edges which can be completed using that edge. If so, it creates a new edge in the current column with the dot advanced and indicies updated. If, as a result, the dot has moved to the end of a rule, the completer also looks back to the column before the starting index of that edge, and sees if there are any other edges that can be updated. It continues this recursively until nothing more can be updated. Finally, if new edges have been added or dots moved, the predictor will recursively predict rules involving nonterminals that are now to the immediate right of a dot. Each column is filled in this way until it reaches the edge of the chart: if the final column of the chart contains a completed S rule, then the parse was successful. Note that to actually derive a parse, it is necessary to maintain pointers which point from completed edges to the two edges which were used to derive them (there may be multiple pairs of edges in the case of ambiguity).

Let us take the simple sentence

John fell off the bike

which can be parsed with the following grammar

S $\rightarrow$ NP VP

NP $\rightarrow$ DT NN

NP $\rightarrow$ NN

VP $\rightarrow$ VB RP NP

DT $\rightarrow$ *the*

NN $\rightarrow$ *bike*

NN $\rightarrow$ *John*

VB $\rightarrow$ *fell*

RP $\rightarrow$ *off*

The chart for the Earley parse of this sentence is below:

		John
Predictor	S $\rightarrow$ $\circ$ NP VP [0,0] NP $\rightarrow$ $\circ$ DT NN [0,0] NP $\rightarrow$ $\circ$ NN [0,0]	VP $\rightarrow$ $\circ$ VB RP NP [1,1]
Scanner		NN $\rightarrow$ *John* $\circ$ [0,1]
Completer		NP $\rightarrow$ NN $\circ$ [0,1] S $\rightarrow$ NP $\circ$ VP [0,1]

After intitial predictions, the scanner reads in *John*, which completes the NP rule, which advances the S dot, and causes the VP rule to be predicted.

Continued overleaf . . .

	fell	off
Predictor		NP → ∘ DT NN [3,3]
		NP → ∘ NN [3,3]
Scanner	VB → *fell* ∘ [1,2]	RP → *off* ∘ [2,3]
Completer	→ VB ∘ RP NP [1,2]	→ VB RP ∘ NP [1,3]

The next two columns involve completing the first two RHS elements of the VP rule, which results in a prediction of an NP rule.

	the	bike
Predictor		
Scanner	DT → *the* ∘ [3,4]	NN → *bike* ∘ [4,5]
Completer	NP → DT ∘ NN [3,4]	NP → DT NN ∘ [3,5]
		VP → VB RP NP ∘ [1,5]
		S → NP VP ∘ [0,1]

the bike completes the NP, which completes the VP, which completes the S.

Continued overleaf . . .

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

- **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, 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) Decoder or the noisy channel model for machine translation (1 mark)
  4. Analysis (1 mark)
     (a) Morphology
     (b) Phrase vs word based

- **Relevance feedback.** User interactions are a key source of feedback in information retrieval. Describe both pseudo- and regular relevance feedback, and outline ways in which relevance feedback can be used in a retrieval system. Describe the strengths and weaknesses of relevance feedback compared with standard one-shot retrieval and supervision through click-through data and query log mining.

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

  1. Clarity (2 marks)
  2. Description
     (a) Query expansion outline, why it is a 2-pass procedure (1 mark)
     (b) Relevance feedback under human supervision (1 mark)
     (c) Pseudo relevance feedback, its assumptions (1 mark)

Continued overleaf . . .

3. Strengths and weaknesses (should mention runtime, obtaining a better query, when it works) (2 marks)
4. Compare to a query log mining (1 mark)

— End of Exam —