Section A - Answer ALL questions

1. Explain briefly why SQL queries are not an example of Information Retrieval.

(2 marks)

Model Answer:

SQL queries are used to obtain information from structured data. IR deals predominantly with unstructured data. (2 marks) *.*

2. Briefly describe how you would evaluate a search engine in terms of relevance of search results to queries.

(4 marks)

Model Answer:

Standard methodology in IR consists of three elements A benchmark document collection (1 mark) *. A benchmark suite of queries* (1 mark) *. An assessment of the relevance of each query-document pair* (1 mark) *. The evaluation would involve comparing the results of the search engine on the benchmark set of queries against the relevant documents.* (1 mark)

3. A user's query text can be a poor proxy for the actual information need he/she is trying to satisfy. Explain why illustrating with a suitable example.

(4 marks)

Model Answer:

An information need is sometimes too abstract and/or too complicated to describe in a few words. (2 marks) *Example: Information need = User needs to find out information about the causes of the latest financial crisis. Query = "causes of the financial crisis". Result = Article on BBC news entitled: "Financial crisis causes a 10% increase in cases of clinical depression."* (1 mark)

Result is almost perfect match for user's query but actually irrelevant to the user's information need. Hence relevance of results to queries cannot be used to measure user satisfaction. (1 mark)

4. Given a search engine and no information about the retrieval model it is based on, how would you determine if it is performing ranked retrieval or boolean retrieval? (Assume multiple query terms are combined by conjunction)

(2 marks)

Model Answer:

I would try a query containing multiple terms A, B, C, etc. If the results include documents that do not contain all the terms, the engine is performing ranked retrieval. (2 marks)

5. You are designing a boolean retrieval search engine for a collection of 8 million documents, each of which is about 1000 words long. Assuming there are about 500,000 unique terms in the collection, calculate the size in bytes of the term-document incidence matrix if we store each element by a single bit.

(1 mark)

Model Answer:

8,000,000 docs x 500,000 words x 1 bit per word per doc= 4,000,000,000,000 bits = 500,000,000,000 bytes = 0.5 Terabytes (1 mark)

Using the above example as illustration, explain why the inverted index form is a much better representation for the term-document incidence matrix.
[Hint: estimate the largest possible size of the inverted index representation]

(3 marks)

Model Answer:

Assuming each document contains 1000 different words, and 4 bytes to identify each of the 8M documents, (1 mark) *8,000,000 docs x 1,000 words x 4 bytes per word per doc= 32,000,000 bytes = 32 MB* (1 mark) *This is much smaller than 0.5 TB which shows why the inverted index representation is much more efficient in terms of memory.*

6. What is the bag-of-words representation of the documents

docA: ”Alice is quicker than Bob who is quicker than Charlie”

docB: ”Charlie is quicker than Alice who is quicker than Bob”

Assume the dictionary is [Alice, Bob, Charlie, is, quicker, than, who].

(1 mark)

Model Answer:

$$v\left(docA\right) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

$$v\left(docB\right) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

(1 mark)

Using your answer as illustration, state the main weakness of the bag-of-words representation

(1 mark)

Model Answer:

The main weakness of the bag-of-words representation is that it ignores word order. In this example, the two documents consist of the same words in different order. Hence they have identical bag-of-words representations despite the fact that their meaning due to word order is very different. (1 mark)

7. What do we mean by **document classification**? Briefly describe TWO applications of document classification.

(3 marks)

Model Answer:

Document classification is the process of automatically determining the class of a document given a set of example documents and their classes. (1 mark)

One application for document classification is a system for automatically determining the categories of email in someone's inbox. The user must specify a set of email messages that he/she considers to belong to certain categories (e.g. spam, business, personal) and then the system automatically determines the class of a new incoming message. (2 marks)

Briefly describe how K-Nearest Neighbour classification works. (2 marks)

Model Answer:

Given a document that we wish to classify, we identify its K nearest neighbours and assign it the majority class in this set of neighbour documents. (2 marks)

If you were given a reliable and efficient ranked-retrieval search engine, how would you turn that into a KNN classifier?

(2 marks)

Model Answer:

Using the document-to-be-classified as a query on the search engine the k nearest neighbours can be efficiently obtained by reading off the top k retrieved results. (2 marks)

8. What are the THREE main types of user need that a search engine satisfies? Give an example for each.

(6 marks)

Model Answer:

Informational: user wants to learn about something. (1 mark) *E.g. ”quantum mechanics”* (1 mark) *Navigational: user wants to go to a particular page.* (1 mark) *E.g ”general motors”* (1 mark) *Transactional: user wants to do something (web-mediated).* (1 mark) *E.g. ”rent a car in rome”*

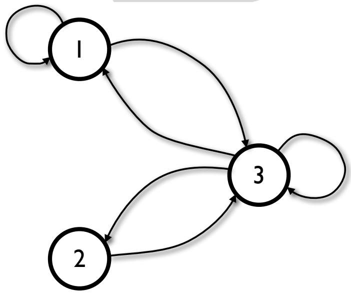
9. Give THREE characteristics of the **world wide web** that make it differ from ordinary document collections (e.g. a collection of legal case documents)

(6 marks)

Model Answer:

Two marks for each. Possibilities are: (1) The web has no design/coordination (2) Distributed content creation, linking, democratization of publishing (3) Content includes truth, lies, obsolete information, contradictions (4) Unstructured (text, html, etc.), semi-structured (XML, annotated photos), structured (Databases) (5) Scale much larger than previous text collections (6) Phenomenal growth

10. This question is about **Pagerank**. The following diagram depicts a small web of three documents (nodes) with links between them (edges).



a) Assuming a random surfer model WITHOUT teleporting, write down the transition probability matrix P.

(3 marks)

Model Answer:

$$P = \begin{bmatrix} 1/2 & 0 & 1/2 \\ 0 & 0 & 1 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

b) If a_1, a_2 and a_3 are the long term visit rates for pages 1, 2 and 3 respectively, write down the system of equations that they need to satisfy. (8 marks)

Model Answer:

$$\begin{aligned} \frac{a_1}{2} + \frac{a_3}{3} &= a_1 \\ \frac{a_3}{3} &= a_2 \\ \frac{a_1}{2} + a_2 + \frac{a_3}{3} &= a_3 \\ a_1 + a_2 + a_3 &= 1 \end{aligned}$$

(8 marks)

c) How would you use the solution of this system of equations for performing ranked retrieval? (2 marks)

Model Answer:

I would be computing the set of documents relevant to a query, say using binary retrieval, and then I would rank those results based on the steady state visit rates (otherwise known as Pagerank scores). (2 marks)

END OF SECTION A **Total: 50 marks**

Section B - Answer any TWO of the following three questions

11. This question is about **tf-idf**. A search engine uses the following term weighting scheme for a term t and a document d:

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \log_{10} \frac{N}{\text{df}_t}.$$

What is signified by the symbols $\text{tf}_{t,d}$, df_t and N? (3 marks)

Model Answer:

Term frequency (1 mark) *, document frequency* (1 mark) *and total number of documents in the collection.* (1 mark)

Identify which part of the formula corresponds to **idf** (1 mark)

.

Model Answer:

$$\textit{idf}_{t,d} = \log_{10} \frac{N}{\textit{df}_t}.$$

(1 mark)

What is the weight given to a term that appears in ALL documents in the collection? (1 mark)

Model Answer:

Zero ($\textit{df}_t = N$). (1 mark)

Using your answer as illustration explain briefly what is the purpose of idf scores. (4 marks)

Model Answer:

Idf puts a penalty on terms that appear in many documents in the collection. For example, when a term appears in all the documents, idf ensures that it gets zero weight. The assumption behind idf is that terms that appear in fewer documents somehow carry more information content. Therefore such terms are given higher scores when we form the vector space representation of a document. (4 marks)

The search engine’s collection contains 1,000,000 documents, two of which are the following:

Doc1: ”digital cameras and video players”

Doc2: ”video cameras and digital video”

Based on this information complete the missing values in the following table:

	Doc1				Doc2			
Dictionary	tf	df	idf	tf-idf	tf	df	idf	tf-idf
and		1,000,000						
cameras						10,000		
digital		100						
video						10,000		
players		10						

(9 marks)

Model Answer:

	Doc1				*Doc2*			
Dictionary	*tf*	*df*	*idf*	*tf-idf*	*tf*	*df*	*idf*	*tf-idf*
and	*1*	*1,000,000*	*0*	*0*	*1*	*1,000,000*	*0*	*0*
cameras	*1*	*10,000*	*2*	*2*	*1*	*10,000*	*2*	*2*
digital	*1*	*100*	*4*	*4*	*1*	*100*	*4*	*4*
video	*1*	*10,000*	*2*	*2*	*2*	*10,000*	*2*	*4*
players	*1*	*10*	*5*	*5*	*0*	*10*	*5*	*0*

(0.25 marks) *for each right answer.*

Using the tf-idf weights you found, calculate the **cosine similarity** between the two documents. Show all steps of your calculation.

(7 marks)

Model Answer:

The unnormalized vectors of tf-idf weights corresponding to the two documents are

$$v_1 = \begin{bmatrix} 0 \\ 2 \\ 4 \\ 2 \\ 5 \end{bmatrix}, \; v_1 = \begin{bmatrix} 0 \\ 2 \\ 4 \\ 4 \\ 0 \end{bmatrix}.$$

(1 mark) *The norms of the two vectors are*

$$\|v_1\| = \sqrt{0^2 + 2^2 + 4^2 + 2^2 + 5^2} = \sqrt{49} = 7$$
$$\|v_2\| = \sqrt{0^2 + 2^2 + 4^2 + 4^2 + 0^2} = \sqrt{36} = 6.$$

(2 marks)

and the cosine similarity is

$$\cos(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \, \|v_2\|} = \frac{0 \times 0 + 2 \times 2 + 4 \times 4 + 2 \times 4 + 5 \times 0}{7 \times 6} = \frac{2}{3}$$

(4 marks)

12. This question is about **content based image retrieval**.

a) Briefly describe traditional image retrieval (non-content-based). (2 marks)

Model Answer:

A human manually labels each image in the collection with a set of textual tags. At query time the user enters a text query that is then matched against the tags in the collection using traditional text retrieval techniques (tf.idf etc).

b) Identify THREE problems with this approach. (3 marks)

Model Answer:

1. Descriptions from different persons will differ

2. Words convey inexact information

3. Not everything can be expressed with words (e.g. mood, visual properties like texture etc)

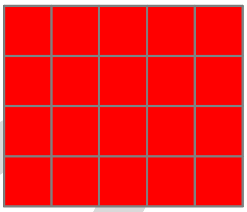
4. Impractical - Too many images

c) Content-based image retrieval is based on **image matching**, i.e. deciding whether or not two images represent the same object, person or scene. Name THREE factors that make image matching a hard problem. (6 marks)

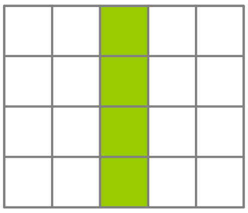
Model Answer:

Two points each. Possibilities include (1) Colour variation due to different pixel sensors, (2) Occlusion, (3) Perspective distortion, (4) lighting, (5) scale and (6) viewpoint variation.

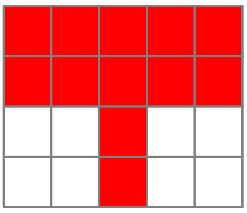
d) Colour Coherence Vectors can be used to perform image retrieval based on colour similarity. Assume we have a collection of the following 7 images shown below.



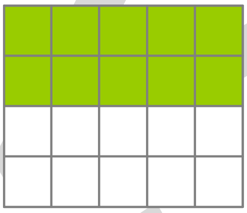
1



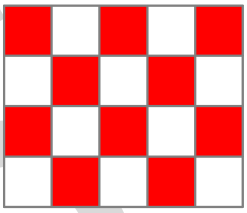
2



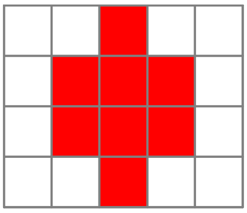
3



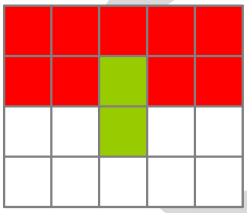
4



5

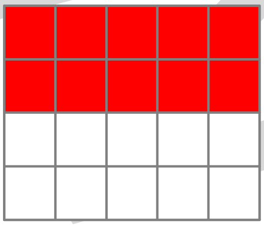


6



7

Further assume we have another image Q (shown below) that we want to use to retrieve similar images in a ”query by example” fashion.



Q

i) Compute the **Colour Coherence Vectors** of all 8 images and the query in numerical form.

(4 marks)

Model Answer:

Numerical form:

1: (20,0,0,0,0,0)

2: (0,0,16,0,4,0)

3: (10,0,8,2,0,0)

4: (0,10,10,0,0,0)

5: (0,0,0,10,0,10)

6: (8,0,8,0,0,4)

7: (9,0,9,0,2,0)

Query: (10,0,10,0,0,0)

ii) Perform image retrieval based on **histogram intersection**. Give all intersection similarity scores and provide the ranking for the collection of images in descending similarity.

(3 marks)

Model Answer:

HI(Q,1) = 10

HI(Q,2) = 10

HI(Q,3) = 18

HI(Q,4) = 10

HI(Q,5) = 0

HI(Q,6) = 16

HI(Q,7) = 18 (2 marks)

Ranking from most to least similar: 3, 7, 6, 1, 2, 4, 5. (1 mark)

iii) By considering the score for image 5, explain how colour coherence vectors improve upon simple colour histogram intersection.

(3 marks)

Model Answer:

Colour histogram cannot capture differences in texture. The best match for Q is under colour histograms only, would be Image 5 purely because the pixel colour frequencies are identical, and despite the fact that 5 is textured while Q is homogeneous. The results are improved in CCV as we are considering two colour histograms per image, one calculated in the homogeneous regions and one in the textured regions.

e) Local Binary Patterns can be used to perform image retrieval based on texture similarity. Assume we are given a grayscale image, part of which is shown below:

12	50	60	19
10	3	50	30
50	10	3	30
50	50	10	5

Compute the Local Binary Pattern for the pixel in the third row and second column.

(2 marks)

Model Answer:

216

If you were using Local Binary Patterns for image retrieval, how would you use the value computed above?

(2 marks)

Model Answer:

When coming across a LBP value of 216 this would increase the 216nd bin of the LBP histogram.

13. This question is about image retrieval using the **bag of visual words** model.

a) Identify and explain TWO desirable properties of a good **keypoint detector**.

(2 marks)

Model Answer:

1. The local image structure around the interest point is rich in terms of local information content

2. It is stable under local and global perturbations in the image domain (due to viewpoint changes, illumination/brightness variations etc) such that the interest points can be reliably computed with high degree of reproducibility.

b) What is a **keypoint descriptor**? (2 marks)

Model Answer:

A keypoint descriptor is a mathematical process for mapping a local image region to a real-valued n-dimensional vector.

c) Identify and explain THREE desirable properties of a good keypoint descriptor.

(3 marks)

Model Answer:

1. Robust: Descriptor is invariant under local and global perturbations in the image domain.

2. Distinctive: Visually different image regions will correspond to dissimilar descriptor vectors.

3. Sparse: The number of elements of the vector is much less than the number of pixels in the image region described.

d) Consider a, somewhat quirky, keypoint descriptor that, for any image region, always outputs a random vector to describe it, that is equal in size to the number of pixels in the image region. Which of the properties you identified above does it satisfy? Justify your answer.

(2 marks)

Model Answer:

This descriptor would be 100% distinctive as every patch gets a completely different descriptor. However it suffers from complete lack of robustness as even the same image region in two different images will be described with entirely different vectors. It is also not sparse as it has the same number of elements as the number of pixels in the image region.

e) **Zero-normalized patches** are a particularly robust keypoint descriptor. Identify TWO image intensity transformations under which this descriptor is invariant.

(2 marks)

Model Answer:

1. Multiplying all pixel intensities in the region by a constant.

2. Adding a constant to all pixel intensities in the region.

f) We would like to create a mobile application for guiding visitors in a museum. Users of this application should be able to obtain information on exhibits around the museum by taking pictures of those exhibits using a hand-held device. Design a simple system for solving this problem using the **bag-of-visual-words** model.

(14 marks)

Model Answer:

In our pre-processing step, we run a keypoint detector and descriptor on each image in our museum collection. We then collect the descriptor vectors from all images and run a clustering algorithm on these vectors (e.g. k-means). Alternatively we use some other type of quantisation algorithm, e.g. k-d trees. The goal is to quantise the vectors so that descriptors corresponding to the same visual object in different images are identified as being the same. (4m)

For each image in the collection we can form a histogram where the bins are the identified descriptor clusters and each bin counts how many keypoint descriptors belong to the corresponding cluster. (2m)

When a query image arrives, this can be converted into the same type of histogram as follows: Run the same keypoint detector and descriptor on the query image. For each descriptor vector, find the nearest cluster (in Euclidean distance). In each bin, count all descriptor vectors that have picked that cluster as the nearest. (4m)

Now that our query image has been reduced to a histogram of quantised descriptors we can use standard text IR techniques like TF.IDF to perform ranked retrieval. The term and document frequencies of TF.IDF will be replaced by the frequencies that the quantised descriptor vectors appear in the collection and query images.(4m)

END OF EXAMINATION PAPER

Total: 125 marks