Web Information Retrieval

Exam Demo (SOSE 2023)
August -, 20-

Prof. Dr. Frank Hopfgartner   Dr. Stefania Zourlidou

Student Name: ______________________________

Matriculation Number: ______________________________

University ID: ______________________________@uni-koblenz.de

Course of Study: □ MSc Web and Data Science

□ Other: ______________________________

Student's Signature: ______________________________

Task:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Sum**
Total Points:	**15**	**15**	**10**	**10**	**10**	**10**	**15**	**15**	**100**
Achieved Points:									

Grade: □ Very Good

□ Good

□ Satisfactory

□ Sufficient

□ Fail

In digits: □

Reviewer's Signature: ______________________________

Please read the following information carefully before solving!

- This exam consists of 8 tasks in 14 pages. Please check your copy for completeness and legibility.
- You will find four additional pages at the end of the exam, which won't be corrected.
- Answer only on the fields, where you supposed to contain the answers.
- Do not forget to write your matriculation ID on every page. Otherwise, the corresponding page won't be corrected.
- Use a non-erasable writing instrument, i.e. ink/ballpoint pen. Do *not* write with a pencil!
- You may use a scientific, non-programmable calculator.

1 Multiple Choice Questions - 15 Points

Indicate for the following questions which options apply to which terms and which do not. Incorrect answers will lead to negative marking. Thus, if you are uncertain about an answer it might be wiser not to answer it. Overall points of this question can not be below zero.

1. Which of the following operations are typically viewed as occurring during the pre-processing stage of a web retrieval system?

□ Yes	□ No	Stemming or Lemmatization
□ Yes	□ No	Relevance Feedback
□ Yes	□ No	Stop word removal
□ Yes	□ No	Anchor text Tokenization

2. Indicate for the following relevancy measures whether a result list would have to be sorted in *increasing* (Inc.) or *decreasing* (Dec.) order on the relevancy measure such that the most relevant result is first, the second most relevant is second, and so forth.:

□ Inc.	□ Dec.	Cosine Similarity
□ Inc.	□ Dec.	Euclidean Distance
□ Inc.	□ Dec.	Jaccard coefficient
□ Inc.	□ Dec.	Kullback-Leibler Divergence

3. Web search is different from the classical form of ad hoc information retrieval. Which of the following properties characterize web search specifically, because they do not apply to the classical model?

□ Yes	□ No	The documents length could be different
□ Yes	□ No	The documents could be spams or duplicates
□ Yes	□ No	The documents are interlinked and connected
□ Yes	□ No	The documents are unstructured

4. Which of the following statements are true about URL frontier of the Mercator scheme?

□ Yes	□ No	The role of Front queue is to ensure politeness and freshness
□ Yes	□ No	Back queue ensures that a particular website is not hit too often
□ Yes	□ No	Heap maintains the distributiveness property of crawler
□ Yes	□ No	Each front queue can have url's from different domains

5. Suppose that A, B, C and D are four different web pages; there exist a link from page A to B, B to C, and C to D. Consider the following distinct scenarios, and decide weather it will *increase* (Inc.) or *decrease* (Dec.) the PageRank score of C.

□ Inc.	□ Dec.	Adding a link from D to B
□ Inc.	□ Dec.	Deleting a link from A to B
□ Inc.	□ Dec.	Adding a link from B to D
□ Inc.	□ Dec.	Adding a link from A to D

2 Evaluation 15 points

Imagine you have an image data collection of 50 images, each belonging to different animal categories: mammal - 20, birds - 10, Fish - 10, amphibians - 10. For a given user query, figure 1 shows the image list presented to the user.

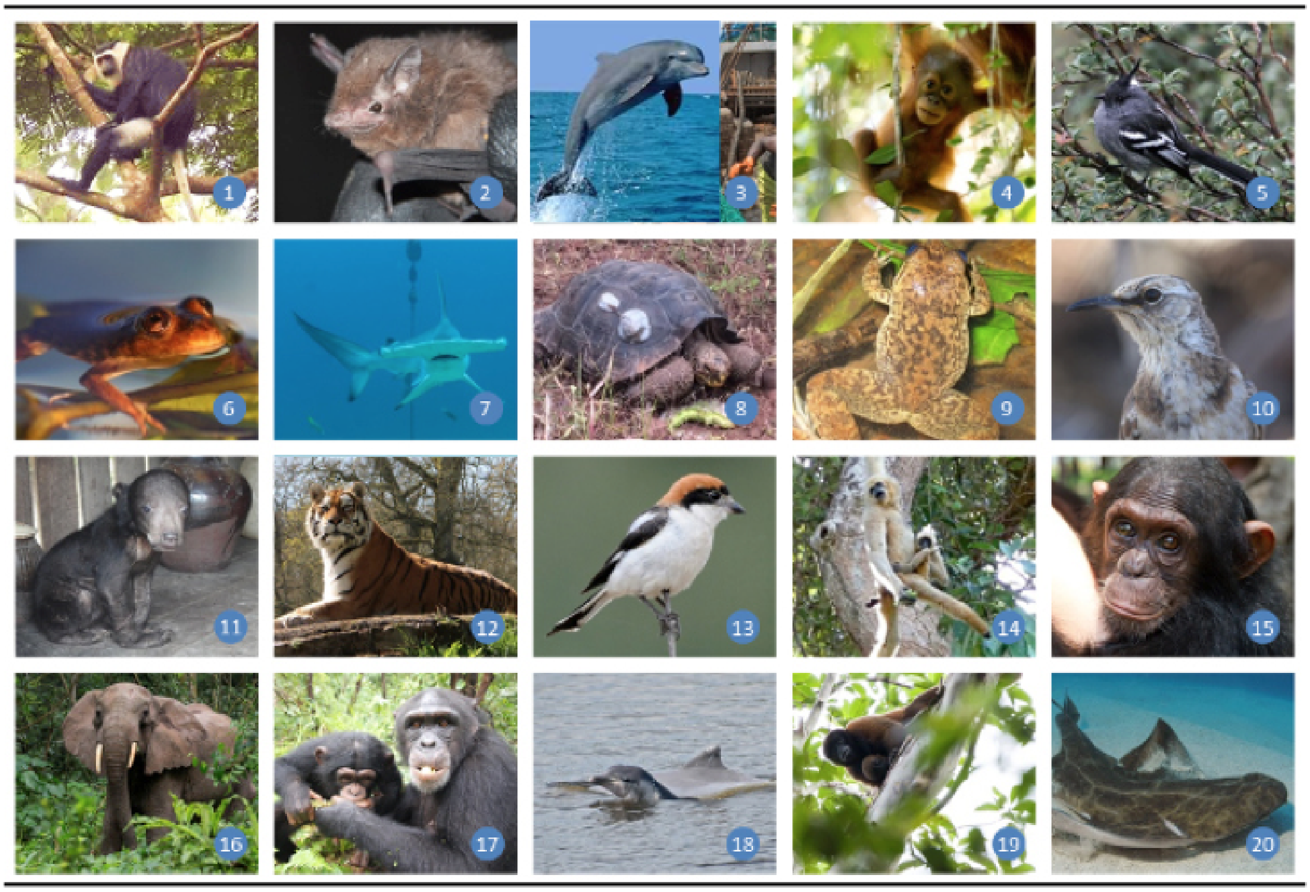


Figure 1: Result Set

1. What would be the the precision and recall of the system, if the user query was “bird” (5 points)?

2. What would be the the precision and recall of the system, if the user query was “animal” (5 points)?

3. What would be the accuracy of the system, if the user's query was "car" (5 points)?

3 Indexing 10 points

3.1 Phrase query 6 points

Given the following corpus:

- D_1: in july sales for home rise
- D_2: increase in home sales in july
- D_3: rise in home sales

Find the results for phrase query "home sales" using the most appropriate indexing structure

1. for: $\longrightarrow D_?(4)$
2. home: $\longrightarrow D_?(5?) \longrightarrow D_?(?) \longrightarrow D_?(?)$
3. in: $\longrightarrow D_?(1) \longrightarrow D_2(2) \longrightarrow D_3(1)$
4. increase: $\longrightarrow D_?(1)$
5. july: $\longrightarrow D_?(2) \longrightarrow D_?(6)$
6. rise: $\longrightarrow D_?(6) \longrightarrow D_?(1)$
7. sales: $\longrightarrow D_?(3) \longrightarrow D_?(4) \longrightarrow D_?(?)$

Query:

1. home: $\longrightarrow D_?(?) \longrightarrow D_?(?) \longrightarrow D_?(?)$
2. sales: $\longrightarrow D_?(?) \longrightarrow D_?(?) \longrightarrow D_?(?)$

Phrase query, position criteria: result = $D_?, D_?$

3.2 Theory 4 points

What is the difference between term-partitioning and document-partitioning index, and why it is used?

4 VSM & Feedback 10 points

You are provided with the following document collection that documents different flavours of ice cream:

- d_1 = `pistachio vanilla vanilla chocolate`
- d_2 = `pistachio chocolate stracciatella chocolate raspberry chocolate`
- d_3 = `stracciatella raspberry greentea raspberry`
- d_4 = `chocolate vanilla tuttifrutti greentea chocolate`
- d_5 = `raspberry stracciatella raspberry`
- q = `stracciatella raspberry chocolate`

1. Specify the TF-IDF vectors for all the documents and query (7 points).

Solution

	d_1	d_2	d_3	d_4	d_5	q	IDF
pistachio							
vanilla							
chocolate							
stracciatella							
raspberry							
greentea							
tuttifrutti							

Table 1: Term Occurance and IDF

	d_1	d_2	d_3	d_4	d_5	q
pistachio						
vanilla						
chocolate						
stracciatella						
raspberry						
greentea						
tuttifrutti						

Table 2: TF * IDF values

$\left\|\vec{d_1}\right\| =$

$\left\|\vec{d_2}\right\| =$

$\left\|\vec{d_3}\right\| =$

$\left\|\vec{d_4}\right\| =$

$\left\|\vec{d_5}\right\| =$

$\|\vec{q}\| =$

2. Calculate cosine similarity between document d_5 and query q vector (3 points).

$\vec{d_5} \approx$

$\vec{q} \approx$

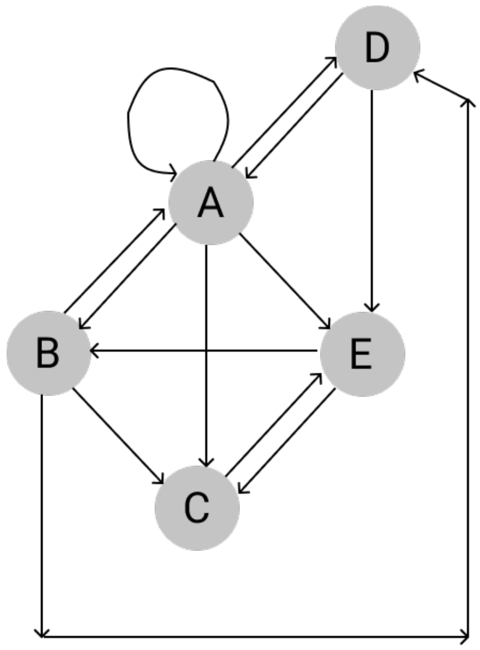
5 Google's Second Price Auction 10 points

Fill the following table for Google's second price auction.

Bid	Clicks Per 1000 Views	Ad Rank	Rank	Paid
$2.00	_____(0.5Pts)	0.04	___(1Pts)	$_______(1Pts)
$3.50	50	______(0.5Pts)	___(1Pts)	$_______(1Pts)
$1.50	_____(0.5Pts)	0.12	___(1Pts)	$_______(1Pts)
$_____(0.5Pts)	10	0.03	4	(minimum)

6 Link Structure 10 points

Consider the following graph. Each node represents a web page and the edges represent the links between them.



1. Calculate that stochastic transition matrix of a random web surfer according to the Page Rank model from your adjacency matrix of the graph using a teleportation rate of α= 0.25.

Solution

1. P=

$$\begin{bmatrix} ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? \end{bmatrix}$$

7 Programming 15 points

You are given following code that performs various functions related to language models. The code contains gaps that have to be filled with box number containing the correct line of code for each gap from the list of choices given below.

Code

```
import pandas as pd

#Calculating term frequency
    def term_frequency():
         tf_df = pd.DataFrame(index = list(word_bag),
columns = ['d0','d1','d2','d3'])
         for i,j in zip(corpus,tf_df.columns):
               for term in word_bag:
                       ________________(1.5Pts)
         return tf_df

#Calculating Probability of term appearing in whole corpus
    def Pmc():
         Total_length = len(d0.split())+len(d1.split())
+len(d2.split())+len(d3.split())
         cf = {} #Count frequency
         P_mc = {}
         for term in word_bag:
                     for doc in _______(1.5Pts):
                             if term in ________(1.5Pts):
                                             _________________(1.5Pts)
                             else:
                                             _________________(1.5Pts)
                     P_mc[term]=cf[term]/Total_length
          return P_mc

#Calculating Probability of term appearing in the document
    def Pmd():
         P_md = tf_df.copy()
         for term in word_bag:
               for j,i in zip(corpus,P_md.columns):
                       P_md[i][term] = ___________________(1.5Pts)
         return P_md


#An Unsmoothed Unigram Model
    def Puni(P_md,q):
         P_uni = {}
         for col in ________(1.5Pts):
              P_uni[col]=1
              for term in q.split():
                      P_uni[col] = _________________(1.5Pts)
```

```
        return P_uni

#A Linear Interpolated Unigram Model
   def Pinterp(P_md,P_mc,q):
       P_interp = {}
       lamb = 0.5
       for col in P_md.columns:
             P_interp[col] = 1
             for term in q.split():
             P_interp[col] = _____________________________(1.5Pts)
       return P_interp

#Main
   d0 = "cats runs behind rats"
   d1 = "dogs runs behind cats"
   d2 = "rats runs cats"
   d3 ="behind runs cats dogs"
   query = "behind rats"

   corpus=[d0,d1,d2,d3]
   word_bag=  ____________________(1.5Pts)

   tf_df = term_frequency()
   pmc = Pmc()
   pmd = Pmd()
   puni = Puni(pmd,query)
   pinterp = Pinterp(pmd,pmc,query)
```

List of Choices

1. ???????
2. ???????
3. ???????
4. ???????
5. ???????
6. ???????
7. ???????)
8. ???????
9. ???????
10. ???????
11. ???????
12. ???????
13. ???????
14. ???????
15. ???????
16. ???????
17. ???????
18. ???????
19. ???????
20. ???????
21. ???????
22. ???????
23. ???????
24. ???????
25. ???????
26. ???????
27. ???????
28. ???????
29. ???????
30. ???????
31. ???????

8 HCI 15 points

How can we design an intuitive and efficient search interface for an e-commerce website that allows users to easily find and compare smartphones based on their preferences and specific technical specifications?

Additional page (it won't be corrected)

Additional page (it won't be corrected)

Additional page (it won't be corrected)

Additional page (it won't be corrected)