Information Retrieval and Recommeder Systems (IRRS)
Master in Data Science (MDS)

# Session 3: Models, part 2

### Exercise List, Fall 2022

---

**Basic comprehension questions.**
**Check that you can answer them before proceeding. Not for credit.**

1. Tell a possible advantage of LSI over the vector model with the cosine measure of similarity. Tell a possible disadvantage.
2. Write down Rochio's formula for user relevance feedback.
3. Explain why the inverted index is adequate for retrieving documents matching a query.
4. Invent a small document collection (5-6 documents with 5-6 words each) and draw the inverted index it would produce.
5. True or false: Query optimization is the process by which one finds the best queries for a given retrieval task.
6. What is the main reason for compressing the index?
7. (Looking at the course slides if you want) Write the self-delimiting unary encoding and Gamma codes of the numbers 17 and 23. Write the Gamma code of 787.

---

**Exercises for credit.** Solving three of these exercises (not solved by the instructors in class) suffice for full credit for this assignment.

## Exercise 1

Consider the following fragment of an inverted index file containing, for each term, the docid's of the documents that contain it and in which positions. For example, document 2 contains "angels" in positions 36, 174, and so on.

**angels**

2: 36, 174, 252, 651

4: 12, 22, 102, 432

7: 7, 17

**fools**

2: 1, 17, 74, 222

4: 8, 78, 108, 458

7: 3, 13, 23, 193

**fear**

2: 87, 704, 722, 901

4: 13, 43, 113, 433

7: 18, 328, 528

**in**

2: 7, 37, 76, 444, 851

4: 11, 20, 110, 470, 500

7: 15, 25, 195

**rush**

2: 2, 66, 75, 321, 702

4: 9, 69, 149, 429, 569

7: 4, 194, 404

**to**

2: 47, 86, 234, 999

4: 14, 24, 774, 944

7: 19, 319, 599, 709

**tread**

2: 57, 94, 333

4: 15, 35, 155

7: 20, 320

**where**

2: 67, 124, 393, 1001

4: 11, 41, 101, 421, 431

7: 16, 36, 736

Which documents satisfy the query “`fools rush in`” `AND` “`angels fear to tread`”?

## Exercise 2

1. Describe an algorithm to create the inverted index term $\rightarrow$ (document,frequence) from a collection of documents, assuming that it is small enough that it all fits in RAM, including the posting lists. We want to keep the posting lists stored by docid.

2. Recall that random accesses to disk are much more expensive than random accesses to RAM and that sequential access to disk. With this in mind, redesign your algorithm so that it has a reasonable performance if the posting lists are too large to fit in RAM and have to be kept in disk. Try to take advantage of all the RAM you have and to use mostly sequential accesses to disk.

## Exercise 3

Refer to the collection in Exercise 5, Topic 2.

1. Which processing order would you recommend for the query `computer AND client AND applications`?

2. Recommend a processing plan for the boolean query

(computing AND programs) OR (p2p AND applications)
OR (computing AND networks AND applications)

## Exercise 4

In a set of $300,000$ documents we have the following term frequencies for some of the existing terms:

| Charles | Dickens | Leon | Tolstoi | Anton | Chejov |
|---|---|---|---|---|---|
| 24.000 | 1.000 | 10.000 | 4.000 | 13.000 | 7.000 |

Propose an evaluation plan for the following query:

(Charles AND Dickens) AND ((Leon AND Tolstoi) OR (Anton AND Chejov))

in order to minimize the list processing time. Justify your answer.

## Exercise 5

1. Given the posting list

$$[10, 1, 15, 3, 22, 2, 23, 4, 34, 1, 44, 1, 50, 2, 58, 8, 90, 1, 101, 1, 112, 2]$$

   (which means that a term appears once in document 10, three times in document 15, two times in document 22, etc.), give the bit string that results of compressing it using self-delimiting unary for the frequencies and gap compression + Elias' Gamma code for the docid's. The first docid in the list (which does not have a gap) is encoded in Gamma code directly.

2. Perform the inverse process on the bit string

$$000010101100010001010001000100011011001000110$$

(Note: since there are dual usages of 0 and 1 in these codes, say that the unary self-delimiting code of 3 is 110 and the Elias Gamma code of 4 is 00100).

## Exercise 6

We create an inverted index from a collection of 1 million documents. We only place 6 terms in the index, with the following frequencies:

| Term | Documents |
|---|---|
| A | 10,000 |
| B | 20,000 |
| C | 40,000 |
| D | 80,000 |
| E | 120,000 |
| F | 150,000 |

Compute the approximate number of comparison between documents needed in order to process the following queries without optimizations 1) in the worst case, and 2) assuming mutual independence between occurrences of terms within documents.

1. ((A and B) and C) and D) and E
2. A and (B and (C and (D and E)))
3. ((A and B) or (C and D)) or (E and F)
4. (A and B) or ((C and D) or (E and F))

5. (A and E) or (B and E)
6. (A and B) or E

## Exercise 7

We have indexed a set of $10^7$ documents. Knowing that terms `A`, `B`, `C`, `D` appear, respectively, in 2 million, 1 million, 800,000, and 20,000 documents, propose an efficient evaluation plan for the boolean query

```
(A and B and C) or (A and B and D)
or (A and C and D) or (B and C and D).
```

Express your plan as a sequence of list intersection and list union instructions. Justify your answer. You do not need to compute the expected cost of your plan.

## Exercise 8

1. You want to compress a sequence of positive natural numbers. Say when you would prefer unary self-delimiting code over Elias' Gamma code, or vice-versa. Give a criterion as precise as possible.
2. If we use Elias' Gamma code in gap compression, what is the largest gap that can be encoded using 1 byte?
3. Give the variable-length encoding of the following posting list:

((1, 3) (7, 1) (19, 5) (35, 4) (52,2))

## Exercise 9

We have a collection of $10^8$ documents. The average document length is 10,000 characters, and the average word length in the documents is 7 characters.

1. Suppose that the collection satisfies Heaps' law in the form $10N^{0.5}$. Estimate the number of different words that you expect to find in the collection.
2. We create an inverted index containing docid's only. Estimate the average length of the posting lists. Hint: Estimate first the number of distinct words per document then the number of total entries in the posting lists, then this.

3. Estimate the average gap in posting lists.

4. We use gap compression + Elias Gamma code to encode the posting lists. Estimate the number of bits that the index will use.

*(Short answers – look only at the order of magnitude 1)* $\simeq 3.5$ *million words 2)* $\simeq 10^4$ *entries per list on average 3) also* $\simeq 10^4$ *4) about* 1 *Terabit* = 125 *Gbytes.)*
If for some of these items you need the result from a previous item that you could not solve, make a reasonable guess and specify clearly "Suppose that the result of item $x-1$ is...".

## Exercise 10

Consider a collection of $D$ documents with an average of $L$ different terms per document, and a total of $T$ different terms among all documents.

1. Suppose that we do not need to keep intradocument frequencies, only whether each terms appear or not in each document, and that we do not use any compression mechanism. How much memory is needed to keep the term-document incidences in a full $D \times T$ matrix form? And as posting lists?

2. Estimate the size of the index (as a function of $D$, $L$, $T$) if we compress it with gap compression and Elias' Gamma code.

3. Imagine now that we do want to keep the intradocument frequencies. Estimate the index size if we compress docid's as in the previous question and the frequencies using self-delimiting unary.

If you need to make assumptions or use reasonable approximations, state them clearly.
*Partial answers: 1)* $T \times D$ *bits and* $\log D \times D \times L$ *bits 2)* $D \times L \times 2 \log_2 \frac{T}{L}$ *bits 3) Assuming Heaps' law e.g.* $20\sqrt{N}$*, then* $T^2/400$ *plus the result in 2).*