CSL 436 Information Retrieval

Winter 2021

Assignment no 1 (Written assignment)

1. n a collection of 3000 documents, 3 documents are relevant to a query q : REL q = {A, B, C} Below are rankings given by three systems for that query q . Only the first 5 doc uments are shown to the user. (Here we mark nonrelevant documents with "N".)

System 1	System 2	System 3
N	N	A
N	A	N
A	N	N
B	N	N
N	B	N

   Which of the three systems would you prefer and why?

2. Consider the terms: • t1 = VNIT • t2 = topper • t3 = Medal
   and the initial query vector q0 = (1, 1, 0). The user gave relevance feedback for three documents:
   Relevant. • d1 = (1, 2, 1)
   • d2 = (3, 2, 1)

   Non-relevant. • d3 = (6, 0, 0)

   Let alpha, beta and gamma be equal to 1. What would be the new query according to Rocchio's update rule? Explain the intuition behind the new query and why it might be better than the query initially given by the user. (Assume that the IR system is using a collection of documents about the VNIT.)

3. Consider the following documents:
   doc1 phone ring person happy person
   doc2 dog pet happy run jump
   doc3 cat purr pet person happy
   doc4 life smile run happy
   doc5 life laugh walk run run
   a) Construct the term–document matrix. Assume that no stemming or stop-word removal is required.
   b) Construct the inverted index required for ranked retrieval for these five documents. Assume that no stemming or stop-word removal is required.

4. How do you compute scores in a complete search system? Explain all the concepts in brief. But in your own words. Do not copy from book or slides. Consider this as an opportunity to learn technical writing.

5. Explain in detail the intuition behind the KMP algorithm.

CSL 436 Information Retrieval

Winter 2021

Assignment no 2(Written assignment)

1. Below is a table showing how two human judges rated the relevance of a set of documents to a particular information need (0 = nonrelevant, 1 = relevant). Let us assume that you've written an IR system that for this query returns the set of documents {2, 5, 6, 7, 8} and assume the documents are ranked (document 2 is 1st document in results, document 5 is 2nd document, etc.

Doc ID	1	2	3	4	5	6	7	8	9	10	11	12
Judge 1	0	0	1	1	1	1	1	1	0	0	0	1
Judge 2	0	0	1	1	0	0	0	0	1	1	1	0

   a. Calculate the precision and recall of your system if a document is considered relevant only if the two judges agree it is relevant.
   b. Calculate the precision and recall of your system if a document is considered relevant if either judge thinks it is relevant.
   c. Calculate the average precision of your IR system based on both relevance scenarios in (a) and (b)
   d. Study what is Kappa measure. Calculate the kappa measure between the two judges.
   e. Is the kappa measure of acceptable value? Can the judgements be used?
2. Why is F1 defined as harmonic mean? What is the relationship between F1 and the break-even point?
3. Assume that we have the following posting list for term
   Term a : <1, 2> <3, 1> <8, 2> <10, 3> <12, 4> <17, 4> <17, 4> <22, 3> <24, 2> <33, 4> <38, 5> <43, 5> <55, 3><64, 2> <68, 4> <72, 5> <75, 5> <88, 2>.
   The posting list indicates that term **a** appears in doc 1 twice and in doc 3 once, etc.

   Assume that we have the following posting list for term
   Term b : <10, 2> <56, 1>

   a) Consider the following conjunctive Boolean query: "term **a** and term **b** ". If no skipping is used how many comparisons do you have to find the intersection of these two lists?

   b) Introduce a skip structure (jumping length=4), draw the corresponding figure then give the number of comparisons involved to process the same query.

   c) State the advantages and disadvantages of large and small skips in the posting list

4. The use of inverse document frequency (IDF) in the vector model of document retrieval can lead to the following anomaly: There can exist documents D and E and collections A and B, where both A and B contain both D and E, and a query Q such that:
   • If Q is posed in the context of collection A, D is judged to be more relevant to Q than E.
   • If Q is posed in the context of collection B, E is judged to be more relevant to Q than D.

   Explain how this can happen.

CSL 436 Information Retrieval
Winter 2021

Programming Assignment 1

Suppose that we are designing a program to simulate the search in a dictionary. Words appear with different frequencies, however and it may be the case that a frequently used word which is in the stop list like "the" appear far from the root if they are sorted lexicographically while a rarely used word such as "consciousness" appears near the root. We want that the words that occur frequently in the text to be placed nearer to the root. Moreover, there may be words in the dictionary for which there is no definition. Organize an optimal binary search tree that simulates the storage and search of words in a dictionary.

Programming Assignment 2(a)

Calculate the weighted minimum edit distance between two strings. Back trace your solution. The confusion matrix for spelling errors is given below.

sub[X, Y] = Substitution of X (incorrect) for Y (correct)

X	Y (correct)																								
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	0	0	7	1	342	0	0	2	118	0	1	0	0	3	76	0	0	1	35	9	9	0	1	0	5	0
b	0	0	9	9	2	2	3	1	0	0	0	5	11	5	0	10	0	0	2	1	0	0	8	0	0	0
c	6	5	0	16	0	9	5	0	0	0	1	0	7	9	1	10	2	5	39	40	1	3	7	1	1	0
d	1	10	13	0	12	0	5	5	0	0	2	3	7	3	0	1	0	43	30	22	0	0	4	0	2	0
e	388	0	3	11	0	2	2	0	89	0	0	3	0	5	93	0	0	14	12	6	15	0	1	0	18	0
f	0	15	0	3	1	0	5	2	0	0	0	3	4	1	0	0	0	6	4	12	0	0	2	0	0	0
g	4	1	11	11	9	2	0	0	0	1	1	3	0	0	2	1	3	5	13	21	0	0	1	0	3	0
h	1	8	0	3	0	0	0	0	0	0	2	0	12	14	2	3	0	3	1	11	0	0	2	0	0	0
i	103	0	0	0	146	0	1	0	0	0	0	6	0	0	49	0	0	0	2	1	47	0	2	1	15	0
j	0	1	1	9	0	0	1	0	0	0	0	2	1	0	0	0	0	0	5	0	0	0	0	0	0	0
k	1	2	8	4	1	1	2	5	0	0	0	0	5	0	2	0	0	0	6	0	0	0	4	0	0	3
l	2	10	1	4	0	4	5	6	13	0	1	0	0	14	2	5	0	11	10	2	0	0	0	0	0	0
m	1	3	7	8	0	2	0	6	0	0	4	4	0	180	0	6	0	0	9	15	13	3	2	2	3	0
n	2	7	6	5	3	0	1	19	1	0	4	35	78	0	0	7	0	28	5	7	0	0	1	2	0	2
o	91	1	1	3	116	0	0	0	25	0	2	0	0	0	0	14	0	2	4	14	39	0	0	0	18	0
p	0	11	1	2	0	6	5	0	2	9	0	2	7	6	15	0	0	1	3	6	0	4	1	0	0	0
q	0	0	1	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	0	14	0	30	12	2	2	8	2	0	5	8	4	20	1	14	0	0	12	22	4	0	0	1	0	0
s	11	8	27	33	35	4	0	1	0	1	0	27	0	6	1	7	0	14	0	15	0	0	5	3	20	1
t	3	4	9	42	7	5	19	5	0	1	0	14	9	5	5	6	0	11	37	0	0	2	19	0	7	6
u	20	0	0	0	44	0	0	0	64	0	0	0	0	2	43	0	0	4	0	0	0	0	2	0	8	0
v	0	0	7	0	0	3	0	0	0	0	0	1	0	0	1	0	0	0	8	3	0	0	0	0	0	0
w	2	2	1	0	1	0	0	2	0	0	1	0	0	0	0	7	0	6	3	3	1	0	0	0	0	0
x	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
y	0	0	2	0	15	0	1	7	15	0	0	0	2	0	6	1	0	7	36	8	5	0	0	1	0	0
z	0	0	0	7	0	0	0	0	0	0	0	7	5	0	0	0	0	2	21	3	0	0	0	0	3	0

Programming Assignment 2(b)

Implement the KMP algorithm.

Programming assignment 3

Implement in-memory array-based version of inverted indices presented in the notes, including all 3 versions of the next method: binary search, sequential search and galloping search. Using your implementation of inverted indices, implement the phrase searching algorithm given in the notes.

To test your phrase searching implementation, you will need a corpus of 256 MB or larger that fits comfortably in main memory of your computer. The English Wikipedia could be a good corpus to use. A subset of that is also fine. Check out how big it is.

Select at least 10,000 phrases of various lengths from the corpus and verify that your implementation successfully locates these phrases. Your selection should include short phrases of length 2-3, longer ones of length 4-10 and very long ones too. Include phrases containing frequent and infrequent terms both. At least half of the phrases should contain at least one very common term, such as an article or a preposition.

Following guidelines from notes, compare the efficiency of phrase searching, using your three versions of the next method. Compute average response times. Plot response times against phrase length for all three versions. For clarity you may need to plot the results for linear scan separately from those of the other two methods.

Select another set of phrases all with length 2. Include phrases containing combinations of frequent and infrequent terms. Plot the response times against L (the length of the longest posting list) for all three versions.

Implement the phrase searching algorithm described below:

```
NextPhrase2 (t1, t2, t3……. tn, position)
        u = next (t1, position)
        v = u
        for  i = 2 to n do
                v = next (ti,v)
        if v= infinity then
                return [ infinity, infinity]
        if v – u = n – 1 then
                return [ u, v]
        else
                return next Phrase2(t1,t2,…..tn, v-n)
```

Repeat the performance measurements using this implementation. Report the results.