**Intelligent Information Retrieval**   
**CSC 575**

**Assignment 1  
Due: Thursday, January 28, 201**6

1. Suppose that we have a standard IR evaluation data set containing 1000 documents. Assume that a particular query in this data set is deemed to be relevant to the following 25 documents in the collection:

**REL = { d1, d5, d6, d10, d88, d150, d200, d210, d250, d300, d400, d405, d450, d472, d500, d501, d530, d545, d590, d600, d635, d700, d720, d800, d900 }**

Two different retrieval systems S1 and S2 are used to retrieve ranked lists of documents from this collection using the above query. The top 10 retrieved documents for these two systems are given below (each list is in decreasing order of relevance).

RET(S1) = d2, d5, d150, d250, d11, d33, d50, d600, d500, d520  
RET(S2) = d250, d400, d150, d210, d999, d3, d501, d800, d205, d300

* 1. Compute the Precision and the Recall graphs for each system as a function of the number of documents returned (for 1 document returned, 2 documents returned, etc). First, compile your calculations in a table which shows the Precision and Recall for S1 and S2 using these query results as a function of the number of documents returned (from 1 to 10). Then, create  graphs comparing Precision and Recall of the two systems.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Precision(S1)** | **Recall(S1)** | **F Measure(S1)** | **Precision(S2)** | **Recall(S2)** | **F Measure(S2)** |
| 1 | 0 | 0 | 0 | 1 | 0.04 | 0.076923077 |
| 2 | 0.5 | 0.04 | 0.074074074 | 1 | 0.08 | 0.148148148 |
| 3 | 0.666666667 | 0.08 | 0.142857143 | 1 | 0.12 | 0.214285714 |
| 4 | 0.75 | 0.12 | 0.206896552 | 1 | 0.16 | 0.275862069 |
| 5 | 0.6 | 0.12 | 0.2 | 0.8 | 0.16 | 0.266666667 |
| 6 | 0.5 | 0.12 | 0.193548387 | 0.666666667 | 0.16 | 0.258064516 |
| 7 | 0.428571429 | 0.12 | 0.1875 | 0.714285714 | 0.2 | 0.3125 |
| 8 | 0.5 | 0.16 | 0.242424242 | 0.75 | 0.24 | 0.363636364 |
| 9 | 0.555555556 | 0.2 | 0.294117647 | 0.666666667 | 0.24 | 0.352941176 |
| 10 | 0.5 | 0.2 | 0.285714286 | 0.7 | 0.28 | 0.4 |

* 1. A single metric that can be used to combine precision and recall is the *F Measure*(see Section 8.3 of the IR Book). Using the F1 measure (F measure with B=1), create graph similar to the above comparing the two systems.
  2. Which system is better? Explain your answer.   
     S2 is better. Both precision and recall of S2 is higher than S1. From the F1 measure graph, it is obvious that S2 is always above S1.

1. Use the query**"jaguar"** on three search engines Google, Yahoo, and Bing. Assume that you are interested in the sense of "Jaguar" as an animal (so a document is relevant if it is about the animal Jaguar). For each of your searches record the top 10 results (do not count the "sponsored" results and only count the main results, not sub-categories of results which are often other documents from the same site as the main result). Make a precision plot for the first 10 returned links as function of the number of returned documents, comparing the precision of the three search engines for this query (use a similar approach as part (b) of problem 1, except only for precision). Which of the search engines did better? Which would have done better if you were interested in "jaguar" as a car? Why do you think there might be a difference among these search engines?

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Google** | | | **Yahoo** | | | **Bing** | | |
|  | **Precision** | **Recall** | **F** | **Precision** | **Recall** | **F** | **Precision** | **Recall** | **F** |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0.5 | 0.1 | 0.1666667 | 0.5 | 0.1 | 0.1666667 | 0 | 0 | 0 |
| 3 | 0.6666667 | 0.2 | 0.3076923 | 0.6666667 | 0.2 | 0.3076923 | 0 | 0 | 0 |
| 4 | 0.5 | 0.2 | 0.2857143 | 0.5 | 0.2 | 0.2857143 | 0 | 0 | 0 |
| 5 | 0.6 | 0.3 | 0.4 | 0.6 | 0.3 | 0.4 | 0.2 | 0.1 | 0.1333333 |
| 6 | 0.5 | 0.3 | 0.375 | 0.5 | 0.3 | 0.375 | 0.3333333 | 0.2 | 0.25 |
| 7 | 0.4285714 | 0.3 | 0.3529412 | 0.4285714 | 0.3 | 0.3529412 | 0.2857143 | 0.2 | 0.2352941 |
| 8 | 0.375 | 0.3 | 0.3333333 | 0.375 | 0.3 | 0.3333333 | 0.25 | 0.2 | 0.2222222 |
| 9 | 0.4444444 | 0.4 | 0.4210526 | 0.3333333 | 0.3 | 0.3157895 | 0.2222222 | 0.2 | 0.2105263 |
| 10 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.2 | 0.2 | 0.2 |

Google did best when finding ‘jaguar’ as animal. And Bing did best if we are searching ‘jaguar’ as car.

The reason for the different results from these search engines may have several factors. They collected different documents/links/pages, used different indexes/algorithms to match the search query, and displayed the results in different order as different measures are weighted.

1. **Inverted Index Construction:**  
     
   Consider the following three short documents:

|  |
| --- |
| **Doc #1**  Glimpse is an indexing and query system that allows for search through a file system or document collection quickly. Glimpse is the default search engine in a larger information retrieval system. It has also been used as part of some web based search engines.  **Doc #2**  The main processes in an retrieval system are document indexing, query processing, query evaluation and relevance feedback. Among these, efficient updating of the index is critical in large scale systems.  **Doc #3**  Clusters are created from short snippets of documents retrieved by web search engines which are as good as clusters created from the full text of web documents. |

* 1. First remove stop words, and punctuation, and apply Porter's stemming algorithm to the 3 documents (**Note:** use the [online stemming application](http://facweb.cs.depaul.edu/mobasher/classes/csc575/porter.html)for this purpose).

|  |  |  |
| --- | --- | --- |
| **Doc #1** | **Doc #2** | **Doc #3** |
| glimps | main | cluster |
| index | process | creat |
| queri | retriev | short |
| system | system | snippet |
| allow | document | document |
| search | index | retriev |
| file | queri | web |
| system | process | search |
| document | queri | engin |
| collect | evalu | good |
| quickli | relev | cluster |
| glimps | feedback | creat |
| default | effici | full |
| search | updat | text |
| engin | index | web |
| larger | critic | document |
| inform | larg |  |
| retriev | scale |  |
| system | system |  |
| us |  |  |
| part |  |  |
| web |  |  |
| base |  |  |
| search |  |  |
| engin |  |  |

* 1. Create an inverted index of the three documents, including the dictionary and the postings. The dictionary should also contain (for each term) statistics such as total number of occurrences in the collection and the document frequency. The postings for each term should contain the document ids and the term frequencies (**depict multiple postings for a term as a linked list, similar to Figure 1.3 in the IR Book**).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Term** | **DocId** | | glimps | 1 | | index | 1 | | queri | 1 | | system | 1 | | allow | 1 | | search | 1 | | file | 1 | | system | 1 | | document | 1 | | collect | 1 | | quickli | 1 | | glimps | 1 | | default | 1 | | search | 1 | | engin | 1 | | larger | 1 | | inform | 1 | | retriev | 1 | | system | 1 | | us | 1 | | part | 1 | | web | 1 | | base | 1 | | search | 1 | | engin | 1 | | main | 2 | | process | 2 | | retriev | 2 | | system | 2 | | document | 2 | | index | 2 | | queri | 2 | | process | 2 | | queri | 2 | | evalu | 2 | | relev | 2 | | feedback | 2 | | effici | 2 | | updat | 2 | | index | 2 | | critic | 2 | | larg | 2 | | scale | 2 | | system | 2 | | cluster | 3 | | creat | 3 | | short | 3 | | snippet | 3 | | document | 3 | | retriev | 3 | | web | 3 | | search | 3 | | engin | 3 | | good | 3 | | cluster | 3 | | creat | 3 | | full | 3 | | text | 3 | | web | 3 | | document | 3 | | |  |  | | --- | --- | | **Term(Sorted)** | **DocId** | | allow | 1 | | base | 1 | | cluster | 3 | | cluster | 3 | | collect | 1 | | creat | 3 | | creat | 3 | | critic | 2 | | default | 1 | | document | 1 | | document | 2 | | document | 3 | | document | 3 | | effici | 2 | | engin | 1 | | engin | 1 | | engin | 3 | | evalu | 2 | | feedback | 2 | | file | 1 | | full | 3 | | glimps | 1 | | glimps | 1 | | good | 3 | | index | 1 | | index | 2 | | index | 2 | | inform | 1 | | larg | 2 | | larger | 1 | | main | 2 | | part | 1 | | process | 2 | | process | 2 | | queri | 1 | | queri | 2 | | queri | 2 | | quickli | 1 | | relev | 2 | | retriev | 1 | | retriev | 2 | | retriev | 3 | | scale | 2 | | search | 1 | | search | 1 | | search | 1 | | search | 3 | | short | 3 | | snippet | 3 | | system | 1 | | system | 1 | | system | 1 | | system | 2 | | system | 2 | | text | 3 | | updat | 2 | | us | 1 | | web | 1 | | web | 3 | | web | 3 | |  |

Dictionary => Posting List

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Term** | **doc. req.** |  |  | | |
| allow | 1 | -> | 1 |  |  |
| base | 1 | -> | 1 |  |  |
| cluster | 1 | -> | 3 |  |  |
| collect | 1 |  | 1 |  |  |
| creat | 1 |  | 3 |  |  |
| critic | 1 |  | 2 |  |  |
| default | 1 |  | 1 |  |  |
| document | 3 |  | 1 | 2 | 3 |
| effici | 1 |  | 2 |  |  |
| engin | 2 |  | 1 | 3 |  |
| evalu | 1 |  | 2 |  |  |
| feedback | 1 |  | 2 |  |  |
| file | 1 |  | 1 |  |  |
| full | 1 |  | 3 |  |  |
| glimps | 1 |  | 1 |  |  |
| good | 1 |  | 3 |  |  |
| index | 2 |  | 1 | 2 |  |
| inform | 1 |  | 1 |  |  |
| larg | 1 |  | 2 |  |  |
| larger | 1 |  | 1 |  |  |
| main | 1 |  | 2 |  |  |
| part | 1 |  | 1 |  |  |
| process | 1 |  | 2 |  |  |
| queri | 2 |  | 1 | 2 |  |
| quickli | 1 |  | 1 |  |  |
| relev | 1 |  | 2 |  |  |
| retriev | 3 |  | 1 | 2 | 3 |
| scale | 1 |  | 2 |  |  |
| search | 2 |  | 1 | 3 |  |
| short | 1 |  | 3 |  |  |
| snippet | 1 |  | 3 |  |  |
| system | 2 |  | 1 | 2 |  |
| text | 1 |  | 3 |  |  |
| updat | 1 |  | 2 |  |  |
| us | 1 |  | 1 |  |  |
| web | 2 |  | 1 | 3 |  |

* 1. What are the search results for the following Boolean queries (in each case explain how you obtained them from the inverted index):
     1. index **AND** query

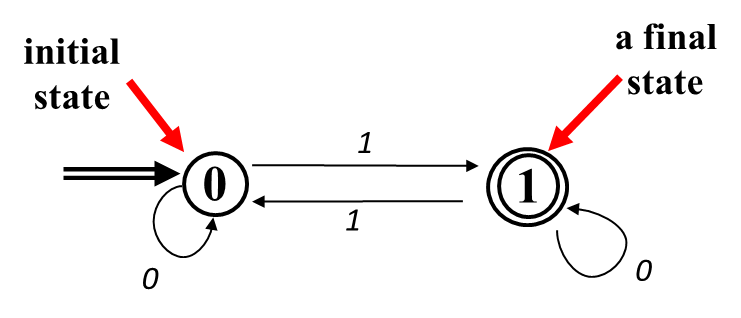
1. Step1: retrieve the postings of ‘index’ => 1, 2
2. Step2: retrieve the postings of ‘queri’ => 1, 2
3. Step3: Merge the two postings => 1, 2
   * 1. index **OR** query
4. Step1: retrieve the postings of ‘index’ => 1, 2
5. Step2: retrieve the postings of ‘queri’ => 1, 2
6. Step3: Find the doc ID in posting list 1 or in list 2 => 1, 2.
   * 1. index **AND** (**NOT** query)
7. Step1: retrieve the postings of ‘index’ => 1, 2
8. Step2: retrieve the postings of ‘queri’ => 1, 2
9. Step3: Find the doc ID in posting list 1 but not in list 2 => NULL
   * 1. (search **AND** query) **OR** (search **AND** retrieve)
10. Step1: retrieve the postings of ‘search’ => 1, 3
11. Step2: retrieve the postings of ‘queri’ => 1, 2
12. Step3: Merge the two postings(first AND) => 1
13. Step4: retrieve the postings of ‘retriev’ => 1, 2, 3
14. Step5: Merge the two postings(second And) => 1, 3
15. Step6: Merge the two postings from step 3 and step 5(Or) => 1, 3
    * 1. (index **OR** cluster) **AND** (web **OR** system)

Suppose, Index => A, cluster => B, web => C, system => D

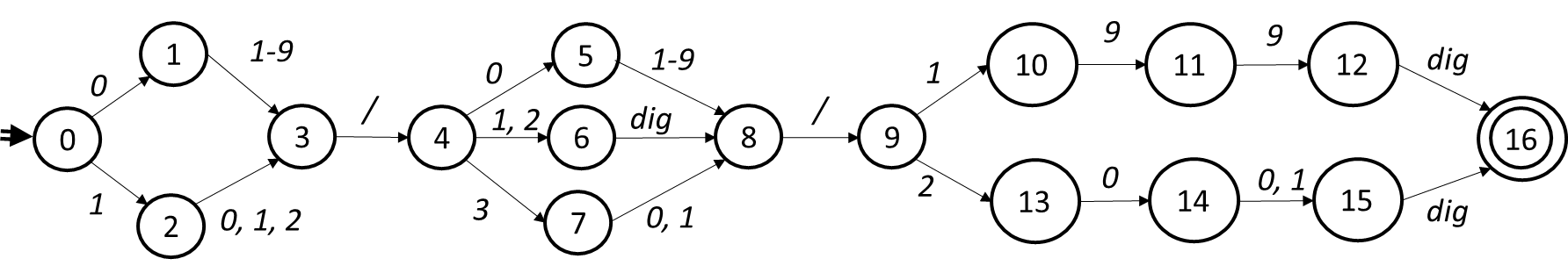
Then the above expression can be converted to (A∨B) ∧ (C∨D), =>((A∨B) ∧ C) ∨((A∨B) ∧ D)=> ((A∧ C) ∨(B ∧ C)) ∨((A∧ D) ∨(B ∧ D))

Calculating the ‘And’ operations first can improve the performance.

1. Step1: retrieve the postings of ‘index’ => 1, 2
2. Step2: retrieve the postings of ‘cluster’ => 3
3. Step3: retrieve the postings of ‘web’ => 1, 3
4. Step4: retrieve the postings of ‘system’ => 1, 2
5. Step5: Merge the two postings(A and C) => 1
6. Step6: Merge the two postings(B and C) => 3
7. Step7: Merge the two postings(A and D) => 1, 2
8. Step8: Merge the two postings(B and D) => Null
9. Step9: Merge above four postings(OR) => 1, 2, 3
10. **Finite State Automata and Lexical Analysis** (**Note:**your FSAs should only accept all terms corresponding to the specified regular expression or pattern, but no other terms).
    1. Construct a Finite State Automata recognizing the set of all binary strings with an odd number of 1's. Examples include: 1, 010, 111, 10110, 0010100111, etc.



* 1. Suppose that during document indexing we are interested in identifying all dates in the format ***mm*/*dd*/*yyyy***, where ***mm*** is a 2 digit string representing the month (ranging from 01 through 12), ***dd*** is a 2-digit string representing the day (01-31), and ***yyyy*** are 4 digits representing the year. Furthermore, we are only interested in the years between 1990 and 2019. **Example: 02/30/2012**. Construct a Finite State Automata that exactly recognizes date tokens as described above. **Notes:** You do not need to match the month with the correct number of days. You may assume that**dig** is a special token representing all digits (0-9). Also, specify a regular expression that corresponds to the finite automata for this language.



Regular expression: (0[1-9]|1[0-2])/(0[1-9]|[1-2]dig|3[0-1])/(199dig|20[0-1]dig)

1. **Ngrams:** Using **bigrams**, which of the following terms is more likely to be considered "equivalent" to the term "**informational"**? Justify your answer by using Dice's coefficient to measure similarity of each term to "**informational**".
   * informal
   * formalization

“informational”

bigrams: in, nf, fo, or, rm, ma, at, ti, io, on, na, al

12 unique bigrams: al, at, fo, in, io, ma, na, nf, on, or, rm, ti

“informal”

bigrams: in, nf, fo, or, rm, ma, al

7 unique bigrams: al, fo, in, ma, nf, or, rm

“formalization”

bigrams: fo, or, rm, ma, al, li, iz, za, at, ti, io, on

12 unique bigrams: al, at, fo, io, iz, li, ma, on, or, rm, ti, za

S(informational, informal) = (2 \* 7) / (12 + 7) = 0.73684

S(informational, formalization) = (2 \* 9) / (12 + 12) = 0.75

So, formalization is more equivalent.

1. The following table shows the raw occurrence frequencies of some words in a hypothetical collection. If ***x*** and ***y*** represent two distinct words, ***f(x)*** and ***f(y)*** denote the raw frequencies for words ***x*** and ***y***, respectively. Assume that the frequency of co-occurrence of ***x*** and ***y*** has already been computed and is denoted by ***f(x,y)***. [Note that the probability of occurrence is estimated as the raw frequency divided by the total number of terms.]

x y f(x) f(y) f(x,y)

-------------------------------------------

color blue 20 40 10

color green 20 10 5

blue green 40 10 5

united states 25 20 15

united airline 25 10 5

states airline 20 10 2

Assuming that the total number of unique words, ***N***,  in the collection is 1000, rank these pairs of words in decreasing order of *expected mutual information*. Show your detailed calculations in at least one case and the final values for all cases.   
We know the formula of mutual information is I(x, y) = . Take ‘color blue’ as example, P(x) = 20/1000, P(y) = 40/1000, P(x,y) = 10/1000, then I(x, y) = = lg12.5/lg2 = 3.64385619

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | y | f(x) | f(y) | f(x, y) | I(x, y) |
| united | states | 25 | 20 | 15 | 4.906890596 |
| color | green | 20 | 10 | 5 | 4.64385619 |
| united | airline | 25 | 10 | 5 | 4.321928095 |
| color | blue | 20 | 40 | 10 | 3.64385619 |
| blue | green | 40 | 10 | 5 | 3.64385619 |
| states | airline | 20 | 10 | 2 | 3.321928095 |

1. [WordNet](http://wordnet.princeton.edu/) is an advanced lexical ontology and thesaurus for the English language with many applications in information retrieval and Web information agents. Write a one page document about WordNet containing the following elements. (1) Your own description of WordNet. (2) What are the unique features of WordNet that distinguish it from other types of standard thesauri? And (3) what are some of the specific ways WordNet can be (or has been) used in the context of information retrieval?

(1). WordNet is a free lexical database of English. It groups the words into sets of synsets based on their meanings. Synonymy is the main relation among words in WordNet. Each synset is linked to others by means of conceptual relation. These relations contains hierarchy structures.

There are only four types of word in WordNet: nouns, verbs, adjectives and adverbs. Determiners, prepositions, pronouns, conjunctions, and particles are not included. Moreover, there are only few adverbs, since most of the adverbs are derived from adjectives.

When searching a word, other words with similar concept will be returned. The search results are divided into four groups: nouns, verbs, adjectives and adverbs, based on their different part of speech.

WordNet gets the definitions from various sources. It setup the word library from ‘Kučera and Francis's Standard Corpus of Present-Day Edited English’ when it was first created. Later, other lists of word were merged to this database. It took several years to collect total 117k synsets until now.

(2). WordNet does not only link the words, but it return synsets with specific meanings. Each has a unique description, then they are semantically disambiguated.

And, WordNet labels words with semantic relations, whereas thesaurus just groups them by meaning similarity without any pattern.

(3).WordNet can be used for computational linguistics and natural language processing. It can also help IR system to analyze the user-input query to understand the real meaning of the search request.