Scores:

1)	6)
2)	7)
3)	8)
4)	9)
5)	10)

COMP 336 Information Retrieval Mid-Term Examination, Spring 2002
April 18, 2002

Time Allowed: 12:10pm to 1:10pm (1 hour)

Name: ______________________ Student ID: __________________

[Note: Answer all questions in the space provided. Answers must be precise and to the point.]

1. [10] Given a document-term matrix of a large collection of documents, describe a method for identifying thesaurus groups (i.e., terms that are used in similar ways within the collection).

   For example, given a document-term matrix of *n* documents with a vocabulary of *t* terms:
   - treat a column as a vector
   - computer the similarity between every pair of columns
   - construct a term-term similarity matrix
   - group term-pairs whose similarity values are large enough

	T_1	T_2	...	T_t
D_1	$d_{1,1}$	$d_{1,2}$	...	$d_{1,t}$
D_2	$d_{2,1}$	$d_{2,2}$	...	$d_{2,t}$
:	:	:		:
D_n	$d_{n,1}$	$d_{n,2}$	...	$d_{n,t}$

2. [5] Explain why stemming can improve both retrieval efficiency and effectiveness.

   Stemming collapses a number of terms into their stem, thus reducing the vocabulary size
   Stemming increases recall by allowing words of the same meaning by different spellings to match (this assumes that the stemming algorithm is accurate enough)

3. [5] (a) Explain why idf cannot be pre-computed for every term.

   idf is based on N and df, which change as documents in the documents are inserted, updated and deleted.

   Note that saying that N is not unknown during insertion time is not correct because pre-computation can be carried out after all documents have been inserted.

   (b) When document scores are computed against a submitted query, cosine similarity measure appears to be expensive to compute because of the complex normalization factor in the formula. Explain how you can speed up the computation of the normalization factor.

   The cosine similarity measure requires the inner product between the document and query vector to be normalized by both the document vector length and query vector length. Several key points are expected from the answer:

   - Document vector length cannot be precomputed when idf is used (see reason above)
   - Avoid using idf in term weights (e.g., use binary weights)
   - If idf is used, precompute document vector lengths and update them periodically (say, after 5% of the collection is updated)
   - Query length is computed only once or can be ignored without affecting the relative ranks of results

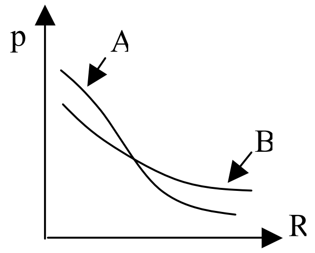
   Important to note here is that cosine similarity (or any other similarity measures) does NOT require a particular term weight formula such as tf.idf to be used. Thus, saying cosine similarity is difficult to compute without saying what term weight method is used is not correct!

4. [10] Give THREE situations under which recall and precision are meaningless or fail to evaluate the effectiveness of an information retrieval system.

   When the precision and recall become meaningless due to division by zero. There are two cases: number of documents returned is zero (precision is undefined) and number of relevant documents is zero (recall is undefined).

   The third problem is that recall and precision will be very high if the number of non-relevant documents for a query is very small (thus requiring fallout rate).

5. [5] Given the precision/recall graphs for two search engines, which one is better as a WWW search engine? Explain why.



For a WWW search engine, users are mostly interested in finding some useful documents and have the patience to only examine the first 2 or 3 pages of results. As such, a high precision at low recall level is more important. That is, A is better.

6. [10] Circle True or False in the following questions:

[✓ T] F A thesaurus can be browsed by a human user so that he/she can formulate better queries to the system.

[✓ T] F The BM pattern matching method is on average faster than KMP method.

[✓ T] F The ideal fallout rate is 0.

T [✓ F] TREC is a popular benchmark for text retrieval system because the document set is small and is carefully selected by experts in the US government.

T [✓ F] The relevance judgment given in TREC is done by human experts examining the relevance between all documents against all queries.

T [✓ F] The BM pattern matching method cannot handle wildcards in a pattern, but KMP method can.

T [✓ F] Precision and recall must add up to 100%

T [✓ F] The inner-product similarity measure in the vector-space model measures the distance between the end points of the query vector and the document vector.

T [✓ F] When stemming is NOT applied to the document terms, it is optional for stemming to be applied to query terms.

[✓ T] F When stopword removal is NOT applied to the document terms, it is optional for stopword removal to be applied to the query.

7. [30] Using the ***basic KMP algorithm***, determine the number of character positions to shift in the shift[] array for the pattern GOOGOL

pattern	shift
G	1
O	1
O	2
G	3
O	3
L	3

In the table given below, perform a step-by-step matching between the text string and the pattern until the entire text string is matched. Note that the number of rows given below is more than the actual number of steps needed by the basic KMP algorithm.

Text string

G	O		G	O	O	G	O	L		G	O	!
G	O	O	G	O	L							
		G	O	O	G	O	L					
			G	O	O	G	O	L				
									G	O	O	G

Note that after a match is found (4th row), the number of shifts is determined by assuming a mismatch at the position marked by the red arrow above. You can see that there is no matching prefix and suffix in the matched string, which is 'GOOGOL'. Thus, the number of shift is equal to the length of the pattern (i.e., 6).

Fill in the shift table below based on the improved KMP method (the one that considers recursive mismatch when the shift table is computed).

pattern	shift
G	1
O	1
O	2
G	4
O	4
L	3

Give the basic delta-1 table in the BM method for the pattern GOOGOL.

mismatch occurs at position:

	1	2	3	4	5	6
!{G, L, O}	1	2	3	4	5	6
G	-	1	2	-	1	2
L	1	2	3	4	5	-
O	1	-	-	1	-	1

8. [5] You are asked to develop a search engine similar to the one you did in your course project. The term weight formula is based on $\frac{tf}{\max(tf)} \times idf$. Describe the *essential* files you have to keep in your index database to support *this weighting formula*, giving the key and the values kept in the file structure. Note that you don't need to show any files or data items to support other operations such as phrase matching, deletion, cosine similarity, etc., and that you don't need to give the exact gdbm syntax.

tf: kept in regular inverted file
max(tf): kept for each document in a forward index
df: can be obtained from regular inverted file
N: keep as a separate variable or count the number of documents in the forward index

Inverted file:
Word ID → document ID, term frequency; …

Forward index:
Document ID → max(tf), word ID, frequency, word ID, frequency, …

Optional:
Wordid → df
DocumentID → max(tf)

9. [10] Fill in the precision, recall and fallout rates in the following table. It is assumed that there are a total of 100 documents, and there are only 3 relevant documents, which are marked with a √ in the first column.

	Rank	doc ID	Recall	Precision	Fallout Rate
	1	1001	0	0	1/97
√	2	2873	1/3	1/2	1/97
√	3	3916	2/3	2/3	1/97
	4	0983	2/3	2/4	2/97
	5	8310	2/3	2/5	3/97
	6	4562	2/3	2/6	4/97
√	7	4921	1	3/7	4/97
	8	7934	1	3/8	5/97
...	. . .	...	...	...	...
	100	3861	1	3/100	97/97

10. [5] Given the following vocabulary obtained from a corpus:

```
police
polices
policing
policy
policies
```

Obtain the successor variety values for the prefixes of the word "polices". Then segment the word using the Peak and plateau method.

p	1
po	1
pol	1
poli	1
polic	3
police	2 or 1
polices	1

Segmented at: polic | es

2 if every word is assumed to end with an invisible end-of-word character. However, it doesn't affect the final segmentation.