IR Assignment I

Figure 1

Doc 1:	breakthrough drug for schizophrenia
Doc 2:	new schizophrenia drug
Doc 3:	new approach for treatment of schizophrenia
Doc 4:	new hopes for treatment of schizophrenia

1. **Draw the term-document incidence matrix for the document collection given in Fig. 1.**

Term\Document	**Doc 1**	**Doc 2**	**Doc 3**	**Doc 4**
approach	0	0	1	0
breakthrough	1	0	0	0
drug	1	1	0	0
for	1	0	1	1
hopes	0	0	0	1
new	0	1	1	1
of	0	0	1	1
schizophrenia	1	1	1	1
treatment	0	0	1	1

2.

A. Draw the inverted index for the document collection given in Fig. 1.

approach	→	1	→	3			
breakthrough	→	1	→	1			
drug	→	2	→	1	2		
for	→	3	→	1	3	4	
hopes	→	1	→	4			
new	→	3	→	2	3	4	
of	→	2	→	3	4		
schizophrenia	→	4	→	1	2	3	4
treatment	→	2	→	3	4		

B. Write the result for the query:

i. schizophrenia AND drug

1111 && 1100 = 1100

schizophrenia AND drug	→	2	→	1	2

ii. for AND not (drug OR approach) assuming both term-document incidence matrix and inverted index representations.

1011 && ! (1100 || 0010) = 1011 && 0001 = 0001

for AND not (drug OR approach)	→	1	→	4

3.

A. **What is indexing granularity? How to resolve the issue related to indexing granularity. Justify.**

Indexing granularity is an issue that arises for very long documents.

For a collection of books, it would usually be a bad idea to index an entire book as a document. A search for *Chinese toys* might bring up a book that mentions *China* in the first chapter and *toys* in the last chapter, but this does not make it relevant to the query.

Instead, we may well wish to index each chapter or paragraph as a mini-document. Matches are then more likely to be relevant, and since the documents are smaller it will be much easier for the user to find the relevant passages in the document. We could also treat individual sentences as mini-documents.

If the units get too small, we are likely to miss important passages because terms were distributed over several mini-documents, while if units are too large we tend to get spurious matches and the relevant information is hard for the user to find.

Indexing granularity is resolved by choosing a suitable size of document unit together with an appropriate way of dividing or aggregating files, if needed.

B. **What are the different ways to determine the vocabulary of terms?**

Different ways of determining the vocabulary of terms include

TOKENIZATION

Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called *tokens*, perhaps at the same time throwing away certain characters, such as punctuation.

DROPPING COMMON TERMS

Sometimes, some extremely common words which would appear to be of little value in helping select documents matching a user need are excluded from the vocabulary entirely. These words are called *stop words*.

NORMALIZATION

Normalization is the process of canonicalizing tokens so that matches occur despite superficial differences in the character sequences of the tokens. The most standard way to normalize is to implicitly create equivalence classes, which are normally named after one member of the set.

STEMMING AND LEMMATIZATION

Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.

Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.

Figure 2

Doc1:	if you prick us do we not bleed
Doc2:	if you tickle us do we not laugh
Doc3:	if you poison us do we not die and
Doc4:	if you wrong us shall we not revenge

4. Draw the complete inverted index for the document collection given in Fig. 2.

and	→	1	→	3			
bleed	→	1	→	1			
die	→	1	→	3			
do	→	3	→	1	2	3	
if	→	4	→	1	2	3	4
laugh	→	1	→	2			
not	→	4	→	1	2	3	4
poison	→	1	→	3			
prick	→	1	→	1			
revenge	→	1	→	4			
shall	→	1	→	4			
tickle	→	1	→	2			
us	→	4	→	1	2	3	4
we	→	4	→	1	2	3	4
wrong	→	1	→	4			
you	→	4	→	1	2	3	4