COMP631 HW1

Yuan Chen

1. Programming

See Attached Python Notebook for details

1.1

```
processed_text
```

```
[['today', 'deepmind', 'released', 'alphacode', 'go', 'alphacod'],
 ['alphacode', 'released', 'today'],
 ['alphacode', 'last', 'week'],
 ['find', 'alphacode', 'news', 'last', 'week'],
 ['alphacode', 'system', 'released', 'deepmind']]
```

1.2

	doc 1	doc 2	doc 3	doc 4	doc 5
today	1	1	0	0	0
deepmind	1	0	0	0	1
released	1	1	0	0	1
alphacode	1	1	1	1	1
go	1	0	0	0	0
alphacod	1	0	0	0	0
last	0	0	1	1	0
week	0	0	1	1	0
find	0	0	0	1	0
news	0	0	0	1	0
system	0	0	0	0	1

1.3

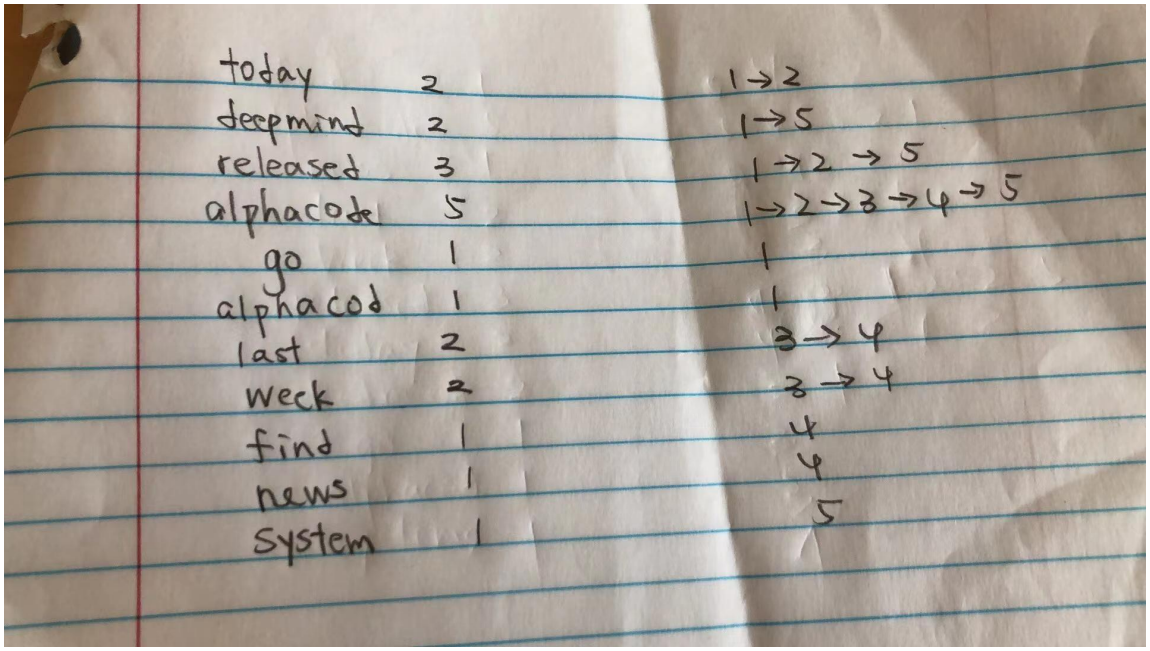
	doc 1	doc 2	doc 3	doc 4	doc 5
today	0.152715	0.305430	0.00000	0.000000	0.000000
deepmind	0.152715	0.000000	0.00000	0.000000	0.229073
released	0.085138	0.170275	0.00000	0.000000	0.127706
alphacode	0.000000	0.000000	0.00000	0.000000	0.000000
go	0.268240	0.000000	0.00000	0.000000	0.000000
alphacod	0.268240	0.000000	0.00000	0.000000	0.000000
last	0.000000	0.000000	0.30543	0.183258	0.000000
week	0.000000	0.000000	0.30543	0.183258	0.000000
find	0.000000	0.000000	0.00000	0.321888	0.000000
news	0.000000	0.000000	0.00000	0.321888	0.000000
system	0.000000	0.000000	0.00000	0.000000	0.402359

1.4

	doc 1	doc 2	doc 3	doc 4	doc 5
doc 1	0.0	0.393133	0.0	0.000000	0.214674
doc 2	0.0	0.000000	0.0	0.000000	0.129474
doc 3	0.0	0.000000	0.0	0.494759	0.000000
doc 4	0.0	0.000000	0.0	0.000000	0.000000
doc 5	0.0	0.000000	0.0	0.000000	0.000000

2. Indexing

2.1 Consider the documents in Question 1, graphically illustrate the inverted index you will create for the documents.



2.2 Given the query “release AlphaCode”, make use of the index we just build to retrieve the matched documents. Illustrate the query processing procedure in details.

1. Locate release in the posting, we get doc 1, doc 2, doc 5
2. Locate alphacode in the posting we get doc1,doc2,doc3,doc4,doc5
3. Merge two list of postings we get doc 1, doc 2 and doc 5
4. Then, we return these three documents

2.3 Explain the advantages of inverted index over term-document incidence matrices for organizing texts?

1. Inverted index is to allow fast full text searches, at a cost of increased processing when a document is added to the database.
2. It is easy to develop.
3. It is the most popular data structure used in document retrieval systems, used on a large scale for example in search engines.

3. Tolerant Retrieval

3. 1) Build a permuterm index for “sydney” and “sidney”, respectively.

3.1.a

sydney$ ydney$s dney$sy ney$syd ey$sydn y$sydne $sydney

3.1.b

sidney$ idney$s dney$si ney$sid ey$sidn y$sidne $sidney

3.2 Consider the wildcard query “s*dney”, explain in details how to search this query in the permuterm indices. What is the rotated wildcard query that we will look up for in the permuterm indices?

We will query dney$s*, which will match dney$sy and dney$si

3.3 Compute the edit distance between the terms in pair as below:

•“AlphaCode” and “AlphaGo” change the c to g and add d, e after the o. Thus, the edit dis3
•“november” and “december” change n, o, v to d, e,c respectively, thus the edit distance is 3
•“condense” and “confidence” add f and i behind first n and change s to c. So edit distance is 3

4) Compute the trigram overlap between each pair of terms list above.

Alp lph pha haC aCo Cod ode
Alp lph pha haG aGo

So 3 trigrams overlap

Trigrams for november are nov, ove, vem, emb, mbe, ber.

Trigrams for december are dec, ece, cem, emb, mbe, ber.

So 3 trigrams overlap

Con, ond,nde,den,ens,nse

Con,onf,nfi,fid,ide,den,enc,nce

So 2 trigrams overlap

4. Evaluation Metrics

4.1 What are the precision and recall of the system on the search?

P = TP/(TP + FP)
= 125/(125 + 50) = 0.71428571428

R = TP/(TP + FN)
= 125/200 = 0.83333333333

4.2 Can you think of a search scenario where recall is preferred over precision? Explain.

When we are screening for cancer, the recall rate is more important because we are more dedicated to finding all possible cases for the patient and saving their lives.

4.3

Accuracy is not an appropriate measure for information retrieval problems for the following reasons

First, in most cases, the data is extremely skewed. A system tuned to maximize accuracy can appear to perform well by simply considering all documents nonrelevant to all queries. Also, trying to label some documents as relevant will lead to a high rate of false positives.

However, labeling all documents as nonrelevant is not useful for information retrieval system users. Users are always going to want to see some documents and can be assumed to have a certain tolerance for seeing some false positives providing that they get some useful information.

The measures of precision and recall concentrate the evaluation on the return of true positives, asking what percentage of the relevant documents have been found and how many false positives have also been returned.