5

CSC415

May 2011

**1ai)** Definition: Cosine similarity helps to determine the closeness of a query vector and each document vector.

Example: Rank the documents based on their close proximity with the query.

**1aii)** Definition: Sorted words commonly have long common prefix. Front-coding is the technique that only stores the differences.

Example: 8automata8automate9automatic10automation

→ 8automat*a1◊e2◊ic3◊ion

**1aiii)** Definition: Information extraction is a type of information retrieval whose goal is to automatically extract structured information from unstructured and/or semi-structured machine-readable documents.

Example: With the following phrases:

*May is studying in NTU.*

*NTU is in Singapore.*

Information extraction can deduce into:

*May is studying in Singapore.*

**1b)** Boolean queries give inclusion or exclusion of documents thus the documents either match or not. It is good for expert users with precise understanding of their needs and the collection. However, Boolean queries often result in either too few or too many results.

On the other hand, in ranked retrieval model, the system returns an ordering over the (top) documents in the collection with respect to a query. Rather than a query language of operators and expressions, which Boolean retrieval use, the user's query is just one or more words in a human language.

**1c)**

| | | | N | | U | | S | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | 0 | 1 | 1 | 2 | 2 | 3 | 3 |
| N | | 1 | 0 | 2 | 2 | 3 | 3 | 4 |
| | | 1 | 2 | 0 | 1 | 1 | 2 | 2 |
| T | | 2 | 2 | 1 | 1 | 2 | 2 | 3 |
| | | 2 | 3 | 1 | 2 | 1 | 2 | 2 |
| U | | 3 | 3 | 2 | 1 | 2 | 2 | 3 |
| | | 3 | 4 | 2 | 3 | 1 | 2 | 2 |

The edit distance between 'NTU' and 'NUS' is 2

**1di)** $\log_2 N$

**1dii)** $\log_2(N/4) + 4$

**1diii)** $\log_2(N/8) + 8$

**2a)**

| | Doc1 | Doc2 | Doc3 | Doc4 |
|---|---|---|---|---|
| **Be** | 1 | 0 | 0 | 1 |
| **Clinton** | 1 | 1 | 0 | 1 |
| **Gore** | 0 | 1 | 1 | 0 |
| **Now** | 0 | 0 | 1 | 2 |
| **President** | 1 | 2 | 1 | 1 |
| **Run** | 1 | 0 | 1 | 0 |
| **Start** | 0 | 0 | 1 | 1 |

**2b)** lnc (Document)

| | Doc1 | | | Doc2 | | | Doc3 | | | Doc4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **tf** | **df** | **n'lize** | **tf** | **df** | **n'lize** | **tf** | **df** | **n'lize** | **tf** | **df** | **n'lize** |
| **Be** | 1 | 1 | 2 | 0 | 1 | 2.449 | 0 | 1 | 2.236 | 1 | 1 | 2.385 |
| **Clinton** | 1 | 1 | 2 | 1 | 1 | 2.449 | 0 | 1 | 2.236 | 1 | 1 | 2.385 |
| **Gore** | 0 | 1 | 2 | 1 | 1 | 2.449 | 1 | 1 | 2.236 | 0 | 1 | 2.385 |
| **Now** | 0 | 1 | 2 | 0 | 1 | 2.449 | 1 | 1 | 2.236 | 1.3 | 1 | 2.385 |
| **President** | 1 | 1 | 2 | 2 | 1 | 2.449 | 1 | 1 | 2.236 | 1 | 1 | 2.385 |
| **Run** | 1 | 1 | 2 | 0 | 1 | 2.449 | 1 | 1 | 2.236 | 0 | 1 | 2.385 |
| **Start** | 0 | 1 | 2 | 0 | 1 | 2.449 | 1 | 1 | 2.236 | 1 | 1 | 2.385 |

ltc (Query)

| | | tf | df | n'lize |
|---|---|---|---|---|
| (i) | **President** | 1 | 0 | 0.301 |
| | **Gore** | 1 | 0.301 | 0.301 |
| (ii) | **Start** | 1 | 0.301 | 0.301 |
| | **Now** | 1 | 0.301 | 0.301 |

<u>**Formula**</u> lnc.ltc (ddd.qqq)

| | Query | Doc1 | |
|---|---|---|---|
| **President** | 1 X 0 X 0.301 = **0** | 1 X 1 X 2 = **2** | 0 X 2 = **0** |
| **Gore** | 1 X 0.301 X 0.301 = **0.091** | 0 X 1 X 2 = **0** | 0.091 X 0 = **0** |
| Similarity | | | 0 + 0 = **0** |

**2bi)** lnc.ltc (ddd.qqq)

| | Query | Doc1 | | Doc2 | | Doc3 | | Doc4 | |
|---|---|---|---|---|---|---|---|---|---|
| **President** | 0 | 2 | 0 | 4.898 | 0 | 2.236 | 0 | 2.385 | 0 |
| **Gore** | 0.091 | 0 | 0 | 2.449 | 0.223 | 2.236 | 0.203 | 0 | 0 |
| Similarity | | | 0 | | 0.223 | | 0.203 | | 0 |

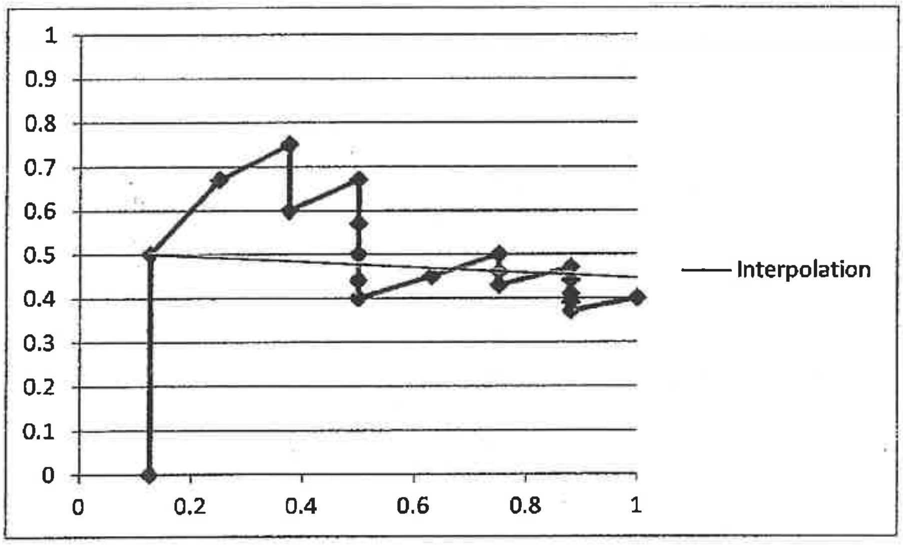
**2bii)** lnc.ltc (ddd.qqq)

| | Query | Doc1 | | Doc2 | | Doc3 | | Doc4 | |
|---|---|---|---|---|---|---|---|---|---|
| **Start** | 0.091 | 0 | 0 | 0 | 0 | 2.236 | 0.203 | 2.385 | 0.217 |
| **Now** | 0.091 | 0 | 0 | 0 | 0 | 2.236 | 0.203 | 3.101 | 0.282 |
| Similarity | | | 0 | | 0 | | 0.406 | | 0.499 |

**2ci)**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **P** | 1 | 0.5 | 0.67 | 0.75 | 0.6 | 0.67 | 0.57 | 0.5 | 0.44 | 0.4 |
| | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| | 0.45 | 0.5 | 0.46 | 0.43 | 0.47 | 0.44 | 0.41 | 0.39 | 0.37 | 0.4 |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **R** | 0.125 | 0.125 | 0.25 | 0.375 | 0.375 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| | 0.63 | 0.75 | 0.75 | 0.75 | 0.88 | 0.88 | 0.88 | 0.88 | 0.88 | 1 |



**2cii)** False. If the first document retrieved is not relevant, Precision will be equal to 0. And if the first document retrieved is relevant, Recall will be more than 0.

**2ciii)** True. Given that N= # docs retrieved, $N_R$ = # relevant docs retrieved, R = # relevant docs in corpus

$P = N_R / N$

$R = N_R / R$

If N = R, P = R

**3ai)**

| **Model ID 1** | **Model ID2** | **Model ID3** | **Model ID4** | **Model ID5** |
|---|---|---|---|---|
| 0.158 fruit | 0.15 need | 0.192 drink | 0.242 fruit | 0.15 time |
| 0.108 or | 0.15 five | 0.192 alcohol | 0.192 and | 0.15 to |
| 0.133 vegetable | 0.2 fruit | 0.192 bed | 0.192 sport | 0.15 eat |
| 0.108 contains | 0.15 day | | | 0.175 vegetable |
| 0.108 enough | | | | |
| 0.108 vitamin | | | | |

**3aii)**

| Model ID | Probability | Rank |
|---|---|---|
| 1 | 0.158 | 4 |
| 2 | 0.2 | 2 |
| 3 | 0.192 | 3 |
| 4 | 0.242 | 1 |
| 5 | 0 | 5 |

**3bi)**

# Vocabulary = 17 (Table Q3a) + 14 (Table Q3b) – 3 (common words) = 28

P(F) = 5/8 and P(S) = 3/8

| | | | |
|---|---|---|---|
| P(fruit \| F) | = (3 + 1) / (20 + 28) | = 4/48 | = 1/12 |
| P(vegetable \| F) | = (2 + 1) / (20 + 28) | = 3/48 | = 1/16 |
| P(sleep \| F) | = (0 + 1) / (20 + 28) | = 1/48 | |
| P(to \| F) | = (1 + 1) / (20 + 28) | = 2/48 | = 1/24 |
| P(enough \| F) | = (1 + 1) / (20 + 28) | = 2/48 | = 1/24 |
| P(bed \| F) | = (1 + 1) / (20 + 28) | = 2/48 | = 1/24 |
| P(OTHERS in F not in S \| F) | = (1 + 1) / (20 + 28) | = 2/48 | = 1/24 |
| P(OTHERS in S not in F \| F) | = (0 + 1) / (20 + 28) | = 1/48 | |

| | | | |
|---|---|---|---|
| P(fruit \| S) | = (0 + 1) / (16 + 28) | = 1/44 | |
| P(vegetable \| S) | = (0 + 1) / (16 + 28) | = 1/44 | |
| P(sleep \| S) | = (3 + 1) / (16 + 28) | = 4/44 | = 1/11 |
| P(to \| S) | = (1 + 1) / (16 + 28) | = 2/44 | = 1/22 |
| P(enough \| S) | = (1 + 1) / (16 + 28) | = 2/44 | = 1/22 |
| P(bed \| S) | = (1 + 1) / (16 + 28) | = 2/44 | = 1/22 |
| P(OTHERS in F not in S \| S) | = (0 + 1) / (16 + 28) | = 1/44 | |
| P(OTHERS in S not in F \| S) | = (1 + 1) / (16 + 28) | = 2/44 | = 1/22 |

**3bii)** Testing: enough fruit

P(F | d) $= 5/8 * (1/24)^1 * (1/12)^1$

$= 0.00217$

P(S | d) $= 3/8 * (1/22)^1 * (1/44)^1$

$= 0.00039$

Classified under <u>Food</u> category

**3biii)** Testing: take action

P(F | d) $= 5/8 * (1/48)^1 * (1/48)^1$

$= 0.00027$

P(S | d) $= 3/8 * (1/44)^1 * (1/44)^1$

$= 0.00019$

Classified under <u>Food</u> category

**3c)** First retrieve all the favorite webpages from the user's bookmarks. Assigns a numerical weighting to each element of a hyperlinked set of documents. A hyperlink to a page counts as a vote of support. The PageRank of a page is defined recursively and depends on the number and PageRank metric of all pages that link to it ("incoming links"). A page that is linked to by many pages with high PageRank receives a high rank itself. If there are no links to a web page there is no support for that page.

For more, read Chapter 17 – Link Analysis.

4a) Micro- and Macro-Averaging is used to combine multiple performance measures into one quantity. Macro-averaging gives equal weight to each class, whereas micro-averaging gives equal weight to each per-document classification decision.

For more, read Chapter 11 – Text Classification 2.

4b) The main function of feature selection in text classification is to improve generalization (performance) by eliminating noise features and avoiding overfitting.

For more, read Chapter 10 – Text Classification 1.

4c)

```
Select $K$ random docs $\{s1, s2, \ldots sK\}$ as seeds.
Until the stopping criterion is met:
    For each doc $di$:
        Assign $di$to the cluster $cj$ such that $dist(xi, sj)$ is minimal.
    (Next, update the seeds to the centroid of each cluster)
    For each cluster $cj$
        $sj = \mu(cj)$
```

The three heuristic methods of choosing the initial seeds are:

- Select good seeds using a heuristic (e.g. doc least similar to any existing mean)
- Try out multiple starting points
- Initialize with the results of another method

For more, read Chapter 12 – Clustering.

4d) The main components of a Web Search Engine include indexes, web crawler and ad indexes.

The purpose of storing an index is to optimize speed and performance in finding relevant documents for a search query. Without an index, the search engine would scan every document in the corpus, which would require considerable time and computing power.

A Web crawler is a computer program that browses the World Wide Web in a methodical, automated manner or in an orderly fashion.

Ad (advertisement) indexes are similar to normal indexes just that it is paid to display its link.

4e) With classification, instead of searching through the whole corpus, we can search only the category that the query belongs to. For example the query "oversea tour", it can be classified under "travel" category. As such, we only need to search documents that are under "travel" category, which can greatly reduce the searching time and improve performance.