Solver: Kenrick

Email Address: kenr0002@e.ntu.edu.sg

1. (a) Information retrieval is the process of retrieving information, based on a specified queries and returned in relevant & ranked human-understandable form. Meanwhile data mining is the process of mining a knowledge out of a raw data, which are not initially apparent, and then present it in human-understandable form.

(b)

(i) Inverted indexes are technique of tracking which documents contain certain terms, by making the terms present in all documents the index; example: [“pen”] → [doc1], [doc2]

(ii) Positional indexes are enhancement of inverted index, where in addition of saving which document contain certain terms, it also remembers the position of that term occurrence within that document; example [“pen”] → [doc1 → [pos1], [pos2]], [doc2 → [pos3]]

(iii) Wild-card queries are type of queries containing asterisks (*) to tell the information retrieval system to replace the asterisks present in the query with any zero or more characters; example: “pen*” will match “pencil” and “pen”

(c) Take note that the substitution cost is 2, meaning that using the table method taught in lectures/tutorials, when comparing the value from diagonal cell, that value is incremented by 2 instead of 1 when the letters are not the same (example below). This is because when comparing the value of diagonal cell, it is meant to check the result of doing “letter substitution” operation in this cell, however, comparing the value of left or top cell is meant for doing “appending a letter” operation.

		b	*a*	*t*	*t*	*l*	*e*
	0	1	2	3	4	5	6
a	1	2	1	2	3	4	5
n	2	3	2	3	4	5	6
t	3	4	3	2	3	4	5

Examples:

- When comparing ‘a’ of “ant” and ‘a’ of “battle”, the value ‘1’ is the minimum of
  - 1 (from 1 + 0 [because ‘a’ == ‘a’]),
  - 3 (from 2 [‘a’] + 1),
  - 3 (from 2 [‘a’] + 1).
- When comparing ‘a’ of “ant” and ‘b’ of “battle”, the value ‘2’ is the minimum of
  - 2 (from 0 + 2 [because ‘a’ != ‘b’]),
  - 2 (from 1 [‘a’] + 1),
  - 2 (from 1 [‘b’] + 1).

Hence, edit distance is 5.

(d) Posting compression works by storing the indexes of postings into a long string of characters, only save the pointer to those words, and by using some technique like blocking or front coding, more memory space can be reduced. The long string in Table Q1a will be “6Antony6Brutus6Ca*esar7◊lpurnia”

(e)

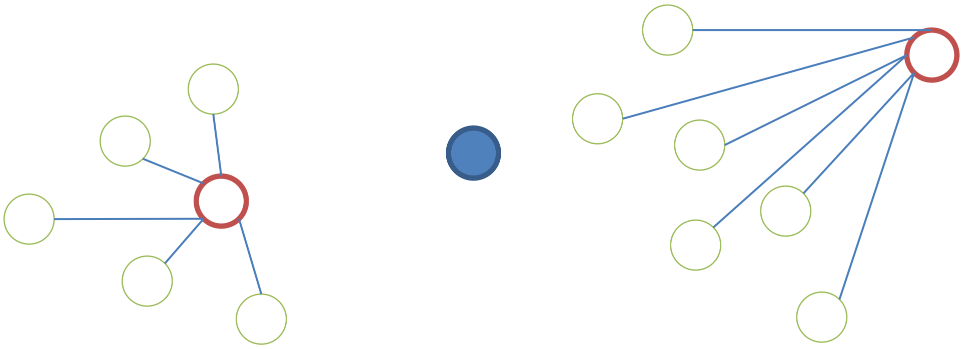
(i)

- q1: “duke of”, “of Cornwall”
- q2: “division of”, “of kingdoms”
- q3: “my lord”, “lord of”, “of Cornwall”

(ii)

- q1: `100 AND 011 = 000`, i.e. None
- q2: `011 AND 110 = 010`, i.e. doc2
- q3: `011 AND 011 AND 011 = 011`, i.e. doc2 and doc3

(f) cluster pruning works by preprocessing: picks sqrt(N) documents at random and call them as leaders, and then for every other documents (followers), compute & save the nearest leaders. When user queries, the IR engine task is to find a nearest leader L and then seek top-k relevant documents under L. Sometimes, this technique might not give the most relevant documents because the initial comparison with the leaders, i.e. when there are other follower documents near the query, but if these followers documents are not under the leader document assigned initially, then they will be ignored. For example,



Here, the query node (solid circle) will be assigned to the left leader node (thick circle) instead of right one (thick circle), but the right one contains more follower nodes (thin circle) that are relevant to the query node, hence the result of the cluster pruning is not correct.

2. (a)

Term	**idf**
Cornwall	$\log_{10}\frac{3}{2} = 0.176$
duke	$\log_{10}\frac{3}{3} = 0$
kingdoms	$\log_{10}\frac{3}{2} = 0.176$
lord	$\log_{10}\frac{3}{2} = 0.176$

(b)

Term	**Document (ntn)**		
	Doc1	**Doc2**	**Doc3**
Cornwall	0	$34 \times 0.176 \times 1$ $= 5.984$	$56 \times 0.176 \times 1$ $= 9.856$
duke	0	0	0
kingdoms	$30 \times 0.176 \times 1$ $= 5.28$	$10 \times 0.176 \times 1$ $= 1.76$	0
lord	$4 \times 0.176 \times 1$ $= 0.704$	$25 \times 0.176 \times 1$ $= 4.4$	0

Term	**Query (atn)**			
	tf	**a**	**t**	**atn**
Cornwall	1	$0.5 + \frac{0.5 \times 1}{1}$ $= 1$	0.176	0.176
duke	0	$0.5 + \frac{0.5 \times 0}{1}$ $= 0.5$	0	0
kingdoms	1	$0.5 + \frac{0.5 \times 1}{1}$ $= 1$	0.176	0.176
lord	1	$0.5 + \frac{0.5 \times 1}{1}$ $= 1$	0.176	0.176

Term	**ntn.atn**		
	Doc1	**Doc2**	**Doc3**
Cornwall	0	5.984×0.176 $= 1.053$	9.856×0.176 $= 1.735$
duke	0	0	0
kingdoms	5.28×0.176 $= 0.929$	1.76×0.176 $= 0.310$	0
lord	0.704×0.176 $= 0.124$	4.4×0.176 $= 0.774$	0

(c)

Term	**ntn.atn**		
	Doc1	**Doc2**	**Doc3**
Cornwall	0	1.053	1.735
duke	0	0	0
kingdoms	0.929	0.310	0
lord	0.124	0.774	0
Cosine similarity	$\frac{0+0+0.929+0.124}{\sqrt{3\times 0.176^2}\times\sqrt{5.28^2+0.704^2}} = 0.648$	$\frac{2.137}{\sqrt{3\times 0.176^2}\times\sqrt{58.26}} = 0.918$	$\frac{1.735}{\sqrt{3\times 0.176^2}\times\sqrt{9.856^2}} = 0.577$
Euclidean distance	$\sqrt{\begin{matrix}0.176^2\\ +0^2\\ +(5.28-0.176)^2\\ +(0.704-0.176)^2\end{matrix}} = 5.134$	7.354	9.68

Using cosine similarity, the rankings are:

1) Doc2
2) Doc1
3) Doc3

Using Euclidean distance, the rankings are:

1) Doc1
2) Doc2
3) Doc3

(d)

Term	**Document (ntc)**			
	Doc1	**Doc2**	**Doc3**	**Query (atc)**
Cornwall	0	$\frac{1.053}{1.343} = 0.784$	1	0.577
duke	0	0	0	0
kingdoms	$\frac{0.929}{0.937} = 0.991$	$\frac{0.310}{1.343} = 0.231$	0	0.577
lord	$\frac{0.124}{0.937} = 0.132$	$\frac{0.774}{1.343} = 0.576$	0	0.577
c	0.937	1.343	1.735	0.3048

Term	**ntc.atc**		
	Doc1	**Doc2**	**Doc3**
Cornwall	0	0.453	0.577
duke	0	0	0
kingdoms	0.572	0.133	0
lord	0.0762	0.333	0
Cosine similarity	0.648	0.919	0.577
Euclidean distance	0.838	0.403	0.919

Cosine similarity did not change (as the values are normalized and the values reflected are the "angle" between the vectors, not the distance), meanwhile Euclidean distance changes as the distances between those vectors are normalized to unit length.

(e)

Term	**ntn.atn**		
	Doc1	**Doc2**	**Doc3**
Cosine similarity	0.618	0.882	0.554
Static quality score	0.49	0.15	x
Net score	1.108	1.032	$0.554 + x$

For doc3 to be the highest ranking,

$$0.554 + x > 1.108$$
$$x > 0.554$$

3. (a) Note: there might be different interpretation of the statements, hence "true" might not be always "true" :P

(i) True

(ii) False

(iii) False

(iv) True

(v) False

(vi) True

(vii) False

(viii) True

(ix) False

(x) False

(b)

	Senior employees	All employees
Consultants	A = 500	C = 14000
Non-consultants	B = 500	D = 21000

N = A + B + C + D = 36000

$$\chi^2 = \frac{\mathrm{N} \times (\mathrm{AD} - \mathrm{CB})^2}{(A+C)(B+D)(A+B)(C+D)} = 40.417$$

Because the value of $\chi^2 > 10.83$, hence the hypothesis "*distribution of senior employees is a reflection of Star Retrieval employee distribution*" is rejected.

(c)

(i) Classification is supervised learning, meanwhile clustering is unsupervised learning. Classification needs training data so the machine knows which category an unseen data belongs to, meanwhile clustering creates grouping of the data based on their feature similarities.

(ii) when clustering the teams based on the projects, soft clustering shall be for junior employees because a person might be working on multiple teams, meanwhile hard clustering is best implemented for senior employees as a person only work on only one team.

(iii) Hierarchical agglomerative clustering (bottom-up) is best suited in this case to group the employees into cluster of teams based on similarity of skill sets; one can use average link to compute the similarity between clusters to solve the issue of chaining issue and outlier issue.

4. (a) Assuming smoothing is being used.

$$Positive: P(pos) \times P(w_1|pos)^3 \times P(w_5|pos) \times P(w_6|pos)$$
$$= \frac{3}{4} \times \left(\frac{5+1}{9+6}\right)^3 \times \frac{0+1}{9+6} \times \frac{0+1}{9+6} = 2.1333 \times 10^{-4}$$

$$Negative: P(neg) \times P(w_1|neg)^3 \times P(w_5|neg) \times P(w_6|neg)$$
$$= \frac{1}{4} \times \left(\frac{1+1}{3+6}\right)^3 \times \frac{1+1}{3+6} \times \frac{1+1}{3+6} = 1.3548 \times 10^{-4}$$

Hence, the prediction is positive.

(b) k-means is an unsupervised learning, where no labelled training data is required; meanwhile kNN is a supervised learning, meaning that labelled training data is required. In this case, there are no labelled training data available, hence the suitable method is k-means.

(c) (i) Euclidean distance is a measure of similarity that is susceptible by the magnitude of the vectors, meanwhile cosine similarity normalizes the vectors and only measure the angle among the vectors.
(ii) $Euclidean\ distance =$
$\sqrt{(0.2-0)^2+(0.8-0.79)^2+(0-0)^2+(0.83-0)^2+(0-0.6)^2} = \sqrt{1.089} = 1.04355$

(d)
These two terms are likely to end up in the same cluster in k-means clustering because a similar set of terms usually will occur in both terms, e.g. both “iPhone 6s” and “Samsung Galaxy S6” usually is accompanied by the terms “phone”, “storage”, “camera”, etc.

(e) (i) “Deep learning” because it is stated that there is a big labelled dataset available. Deep learning is a supervised (meaning that it requires labelled training data) machine learning technique that has recently got very popular due to its accuracy. For doing satisfaction measurement, maybe by having a big labelled dataset will help the machine understand sarcasm, which couldn’t be detected if one uses keyword spotting.
(ii) Both stemming and lemmatization is a normalization process to standardize words occurring the documents. Stemming is a crude chopping letters off words using simple rules that may result in a word that does not exist in the common dictionary, meanwhile lemmatization turns the word into its base form using a thesaurus. For example, “mining” is lemmatized to “mine”, meanwhile it may be stemmed to “min”

My personal course “cheatsheets” can be obtained at: blog.kenrick95.org/resources

For reporting of errors and errata, please visit pypdiscuss.appspot.com
Thank you and all the best for your exams! ☺