Qn I. [In the following, you must SHOW YOUR WORK to get partial credit] Assume that the total number of documents in a corpus is 1024 and that the following words occur in the following number of documents:

"Computer" occurs in 32 documents — 2^5
"software" occurs in 8 documents — 2^3
"intelligent" occurs in 16 documents — 2^4
"robust" occurs in 1024 documents — 2^{10}

1. [6pt] Calculate the TF-IDF weighted term vector for the following document D. Assume that the log in the idf weight is taken to the base 2. (Hint: all the numbers above are powers of 2).

"Computer intelligent software robust computer software"

tf x idf

$$w(\text{Computer}) = 2 \times \log_2\left(\frac{2^{10}}{2^5}\right) = 2 \times 5 = 10$$

$$w(\text{Software}) = 2 \times \log\left(\frac{2^{10}}{2^3}\right) = 2 \times 7 = 14$$

$$w(\text{intelligent}) = 1 \times \log\left(\frac{2^{10}}{2^4}\right) = 2 \times 6 = 6$$

$$w(\text{robust}) = 1 \times \log\left(\frac{2^{10}}{2^{10}}\right) = 1 \times 0 = 0$$

$$D = \begin{array}{cccc} c & s & i & r \\ \end{array}$$
$$D = [10, 14, 6, 0]$$

c, s, i, r are the dimensions of the vector

Corresponding to
Computer
Software
intelligent
Retrieval

R

2.[4pt]Suppose I have a query Q which is specified as
"Intelligent Software"

Assuming that query vector is computed just in terms of TF weights (no IDF weights), and similarity is measured by the cosine metric, what is the similarity between Q and D?

$$\begin{array}{cccc} c & s & i & r \end{array}$$

$$Q = [0 \quad 1 \quad 1 \quad 0]$$

$$D = [10 \quad 14 \quad 6 \quad 0]$$

Important

Q & D must be aligned

So weight for random

$$\text{Sim} = \frac{0\cdot 10 + 1\cdot 14 + 1\cdot 6 + 0\cdot 0}{\sqrt{2}\,\sqrt{10^2+14^2+6^2}} = \frac{20}{\sqrt{2}\,\sqrt{332}}$$

3.[3pt] Suppose the user is shown D in response to the query Q, and the user says that D is relevant to his query. If we now use relevance feedback to modify Q, what will the query vector become? Assume that alpha, beta and gamma are all 1.

$$Q' = \underset{1}{\alpha}\, Q + \underset{1}{\beta}\,(\text{centroid of relevant docs}) - \underset{1}{\gamma}\,(\text{centroid of irrelevant docs})$$

D, since only one doc

no irrelevant docs

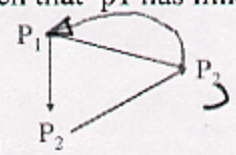
$$= [0 \quad 1 \quad 1 \quad 0] + [10 \quad 14 \quad 6 \quad 0] + 0$$

$$\begin{array}{cccc} c & s & i & r \end{array}$$

$$= [10 \quad 15 \quad 7 \quad 0]$$

R

Qn II. Suppose we have 3 web pages p1, p2 and p3, such that p1 has links to p2 and p3;



p2 has link to p3 and p3 has link to p1 (see the picture)

(a) [6pt] Show one iteration of authorities and hubs algorithm. Assume you set all the authorities and hub values to 1 in the beginning. Show all the steps.

$$A = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \end{array} \begin{array}{c} \begin{array}{ccc} P_1 & P_2 & P_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{array}$$

a_{ij} is 1 if P_i has a link to P_j

$$Au_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$H_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$Au_1 = A' \times Au_0 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$$

$$\widehat{Au_1}^{\text{normalized}} = \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$$

$$Hu_1 = A \times Au_1 = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1/\sqrt{6} \\ 1/\sqrt{6} \\ 2/\sqrt{6} \end{bmatrix} = \begin{bmatrix} 3/\sqrt{6} \\ 2/\sqrt{6} \\ 1/\sqrt{6} \end{bmatrix}$$

$$\widehat{Hu_1}^{\text{normalized}} = \frac{1}{\underbrace{\sqrt{\frac{9}{6} + 4/6 + 1/6}}_{\phi\ 1/\sqrt{14/6}}} \begin{bmatrix} 3/\sqrt{6} \\ 2/\sqrt{6} \\ 1/\sqrt{6} \end{bmatrix} = \begin{bmatrix} 3/\sqrt{14} \\ 2/\sqrt{14} \\ 1/\sqrt{14} \end{bmatrix}$$

(b) [5pt] Show the augmented transition matrix, that will be used by the PageRank algorithm , assuming that with **c** probability a random surfer will follow the links on the current page, and that with **(1-c)** probability she will transition to any of the (three) pages with uniform probability; where **c** is set to 0.8

$$0.8 * \begin{bmatrix} 0 & 0 & 1 \\ 1/2 & 0 & 0 \\ 1/2 & 1 & 0 \end{bmatrix} + 0.2 * \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

from P_1 the surfer transitions to P_2 or P_3 with equal prob

$$= \begin{bmatrix} 0 & 0 & 0.8 \\ 0.4 & 0 & 0 \\ 0.4 & 0.8 & 0 \end{bmatrix} + \begin{bmatrix} 0.066 & 0.066 & 0.066 \\ 0.066 & 0.066 & 0.066 \\ 0.066 & 0.066 & 0.066 \end{bmatrix} = \begin{bmatrix} 0.066 \end{bmatrix}$$

$$= \begin{bmatrix} 0.066 & 0.066 & 0.866 \\ 0.466 & 0.066 & 0.066 \\ 0.466 & 0.866 & 0.066 \end{bmatrix}$$

(c) [2pt] Suppose we set **c** to 0, then what will be the page ranks associated with the three pages?

If $c = 0$, then we have just uniform transition probabilities from every page to every other page

So the Pagerank, which is just the stationary probability distribution is going to be 0.333 for all pages

in contrast for the M^* in part a, the page ranks will be

0.384 P_1
0.220 P_2
0.396 P_3

A note on interpreting eigenvectors as probability distributions — you need to divide each element of the vector by the sum of all elts (so it all sums to 1)

Qn III. Consider the following T-D matrix defining 6 documents defined in terms of 4 keywords.

	K-B	K-B	B-I	B-I	B-I	
	D1	D2	D3	D4	D5	D6
Bush	5	15	7	9	7	0
Kalahari	5	7	1	0	1	0
Iraq	1	0	7	4	6	0
Saddam	0	1	6	4	0	4

We decide to reduce the noise and dimensionality of this data through SVD analysis. The SVD of this T-D matrix, according to MATLAB is: USV^t where U,S,V are given by:

only first two factors

U = T-f

0.8817	0.1969	-0.0444	-0.4264
0.2887	0.4928	0.1190	0.8122
0.3033	-0.6652	-0.5674	0.3790
0.2173	-0.5253	0.8136	0.1222

S = f-f

23.33	0	0	0	0	0
0	9.76	0	0	0	0
0	0	5.03	0	0	0
0	0	0	3.27	0	0

Intuitively the docs are either about Kalahari Bushmen of Africa (D_1, D_2) or about Bush and Iraq war (D_3, D_4, D_5). (See below for D_6)
D_6 is really about Saddam and is thus really similar to Bush-Iraq docs.

V^t = f-d

d1	d2	d3	d4	d5	d6
0.2638	0.6627	0.4237	0.4293	0.3549	0.0373
0.2850	0.6018	-0.6079	-0.3061	-0.2171	-0.2151
-0.0385	0.1948	0.1425	0.1162	-0.7138	0.6460
0.7038	-0.1795	0.3700	-0.5590	0.0308	0.1491
0.5557	-0.3294	-0.1526	0.6077	-0.3198	-0.2965
-0.2090	0.1411	0.5201	-0.1635	-0.4629	-0.6519

only first two factors taken

(1) [3pt] Suppose we are willing to sacrifice upto a maximum of 10% of the total variance in the data, then what is the least number of dimensions we need to keep? Explain how you arrived at your answer.

If we take one dimension the loss is $\left(1 - \frac{23.33^2}{23.33^2 + 9.76^2 + 5.03^2 + 3.27^2}\right)$
$= 0.194$ or 19.4%.

If we take two dimensions the loss is $\left(1 - \frac{23.33^2 + 9.76^2}{23.33^2 + 9.76^2 + 5.03^2 + 3.27^2}\right)$
$= 0.05331$ or 5.3%. Since we want less than 10% loss, we need 2 dims

(2) [4pt] Suppose we decided to just keep top two most important dimensions after the LSI analysis. Draw a bounding box around the parts of U,S,V matrices above that will be retained after this decision. [You answer this question by directly marking the matrices above]

$\boxed{\text{See above}}$

with 2-D the reconstructed data looks as follows

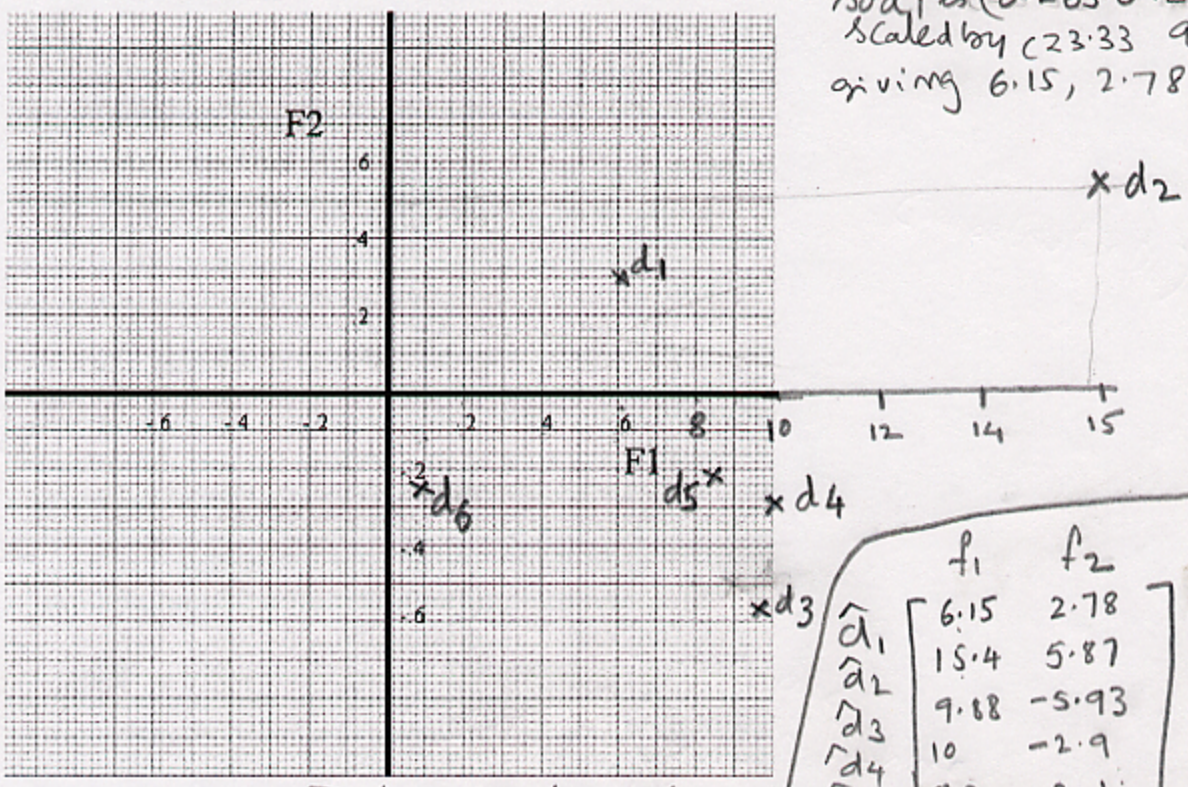
$$\begin{bmatrix} 5.97 & 14.79 & 7.54 & 8.2 & 6.9 & 0.3 \\ 3.14 & 7.36 & -0.07 & 1.4 & 1.3 & -0.8 \\ 0.01 & 0.77 & 6.94 & 5.0 & 63.9 & 1.6 \\ -0.12 & 0.27 & 5.26 & 3.7 & 2.9 & 1.29 \end{bmatrix}$$

Compare this to original T-D Matrix

to do 3 correctly, we need to note that the coordinates of d_i are given by $[f\text{-}f \times f\text{-}d]$ (ie, scale f-d coord by the singular values.

(3) [6pt] Suppose the two most important dimensions after LSI are called *f1* and *f2* respectively. Plot the six documents as points in the factor space (use the plot below). (It is okay if you put the points in the rough place they will come; no need to spoil your eyesight counting all the small grid lines). *Comment on the way the documents appear in the plot—is their placement related in any rational way to their similarity you would intuitively attach to them?*

so d_1 is (0.263 0.285) scaled by (23.33 9.76) giving 6.15, 2.78



Also, d_6 has low angle to d_3, d_4, d_5 but to d_1, d_2, showing it is about Bush-Iraq subject

notice that d_1, d_2 vectors have very low angle (highly similar), while d_3, d_4, d_5 have very low angle. So it does make sense

	f_1	f_2
$\hat{d}_1$	6.15	2.78
$\hat{d}_2$	15.4	5.87
$\hat{d}_3$	9.88	-5.93
$\hat{d}_4$	10	-2.9
$\hat{d}_5$	8.3	-2.1
$\hat{d}_6$	0.8	-2.1

here are all coordinates

(4) [5] What is the vector space similarity between D5 and D6 *before* and *after* the LSI transformation (assume, in the latter case, that we are using the top two dimensions). Is the change intuitively justified?

before

$$\frac{[7\ 1\ 60]\cdot[0\ 0\ 0\ 4]}{\text{normalizing factor}}$$

$= 0$

(Since d_5, d_6 have no common terms, their similarity is zero)

after

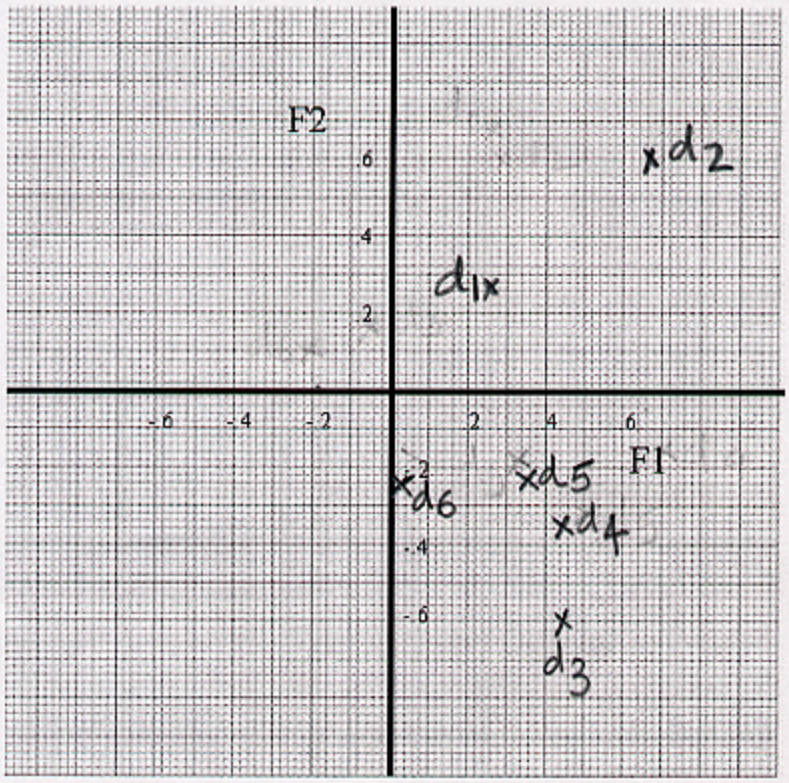
$$\frac{(8.3\ \ -2.1)\cdot(0.8\ \ -2.1)}{|(8.3, -2.1)|\ |(0.8, -2.1)|}$$

$$= \frac{11.05}{8.56 \times 2.24} = \underline{\underline{0.57}}$$

pretty high similarity. LSI allows us to see that the docs are similar even though they dont share words

(wrong answer — Post given for reference)

(3) [6pt] Suppose the two most important dimensions after LSI are called *f1* and *f2* respectively. Plot the six documents as points in the factor space (use the plot below). (It is okay if you put the points in the rough place they will come; no need to spoil your eyesight counting all the small grid lines). *Comment on the way the documents appear in the plot—is their placement related in any rational way to their similarity you would intuitively attach to them?*



[This is what the plot looks like if you plotted f-d without the scaling with f-f

The other plot essentially elongates x axis 2 times more than y axis (23.3 vs 9.76). stretched

(had the ratio between λ_1 & λ_2 been much higher, things would start looking way different in the two plots)

(4) [5] What is the vector space similarity between D5 and D6 *before* and *after* the LSI transformation (assume, in the latter case, that we are using the top two dimensions). Is the change intuitively justified?

before

$$\frac{[7\ 1\ 6\ 0]\cdot[0\ 0\ 0\ 4]}{|D_5|\ |D_6|} = 0$$

no common words between D5 & D6 so no similarity

after If we use f-d without f-f scaling

$$\frac{(0.35\ \ -0.21)\cdot(0.03\ \ -0.21)}{\sqrt{.35^2+.21^2}\ \sqrt{0.03^2+0.21^2}} = 0.627$$

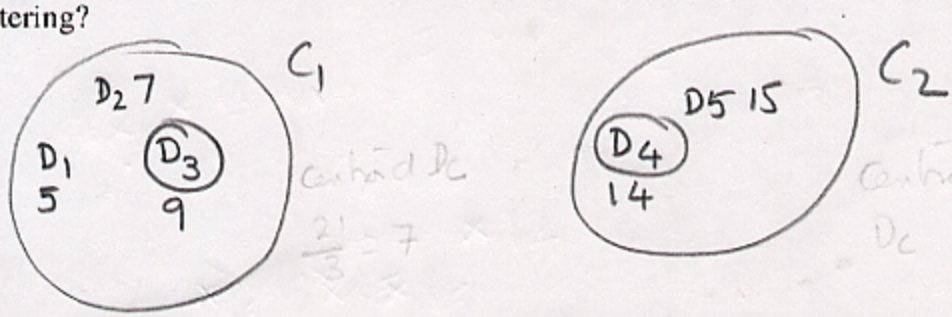
Qn IV Suppose you have a set of documents that basically contain only one key word, repeated multiple times. Suppose the dissimilarity between the documents is judged in terms of the difference in frequency of occurrence of that single keyword. The documents are given by:

D1: 5
D2: 7
D3: 9
D4: 14
D5: 15

(a)[6pt] Suppose we want to use K-Means algorithm to cluster this data into 2 clusters. Show how the clustering progresses, if you start K-means off with D3 and D4 as the seeds. What is the cumulative intra-cluster dissimilarity measure for the final clustering?

Iteration 1

Each doc goes to the cluster whose center it is closest



Iteration 2 Center of C_1 $\frac{9+7+5}{3} = \frac{21}{3} = 7$

Center of C_2 = 14.5

Re-Cluster with 7 & 14.5 as the cluster centers

No change. So stop

Dissimilarity = $(5-7)^2 + (7-7)^2 + (9-7)^2 + (14-14.5)^2 + (15-14.5)^2$

8.5

(b)[2pt] Suppose we are allowed vary $\bar{K}$, the number of clusters that K-means looks for. What is the lowest intra-cluster dissimilarity measure that can be achieved this way? When will it happen?

As we discussed in class, if we care only about intra cluster dissimilarity and are allowed to look for any # of clusters, then the best is each element in its own cluster.

So 5 clusters. Dissimilarity = 0 !

Qn V. Short answer questions. Except for the first question, all other questions carry 3 points.

1. [5pt] Suppose the number of keywords (size of vocabulary) is V, the average length of a document (in terms of words) is N, the number of documents in the corpus is M, the average length of a query is Q, and the average number of documents in which a query word appears is B. What is the time complexity, in vector-space retrieval, of: (a) Naïve query processing (without inverted index) and (b) query processing with inverted index. Why is b better?

Naive
for each keyword, you look at each doc (and the query)

$V \times M$

#Keywords # docs.

Inverted index
for each query word, you just look at docs that contain it

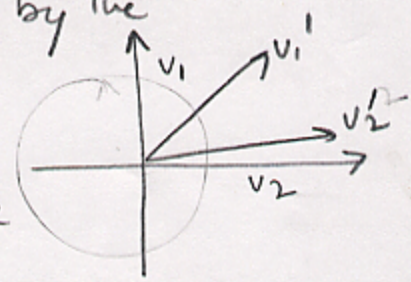
$Q \times B$

words in query # docs in which a query word appears

$QB \ll VM$
since $Q \ll V$; $B \ll M$ in general.

2. [3] In the class, I mentioned that one way of making A/H computation more stable is to define the page importance in terms of subspaces rather than eigen vectors of the adjacency matrix. Explain how/why this is supposed to help. (Short answer in terms of examples is enough)

The idea is that subspaces spanned by the eigen vectors may remain stable w.r.t changes to the (adjacency) matrix even though the individual eigen vectors may change quite drastically.



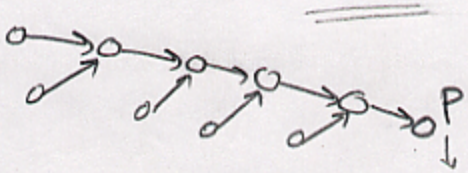
In the figure on the right, both v_1, v_2 and v_1', v_2' span the same subspace (the plane of this paper).

Paper
Efficient
Computation of
Page rank

Interestingly Haveliwalah suggests removing such sink nodes (he calls them dangling nodes) for efficiency. It basically makes no sense to do it later

3. [3] In the class, someone asked why Google doesn't remove all sink nodes (i.e., nodes that do not have any outlinks) from the page graph altogether before computing the page rank. What useful capability of Google will be lost if this were to be done?

If you do that, you may be removing important pages (see P in the picture on left) that have probably not yet been crawled.



can be whitehouse

(Google can return a page that it never crawled — see the white house URL in Google Paper)

4. [3] We talked about "stemming" as a technique that many text retrieval systems use. Comment on how stemming affects the precision and recall (i.e. improve/worsen)

Stemming increases recall, but can reduce precision (some of the returned pages may be less relevant.)

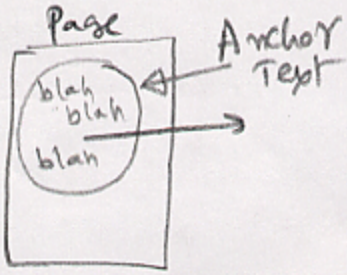
Eg Query = "hostages killed" Stemmed Query: "hostage kill"

But I am looking for

Google basically does not use stemming

5. [3] The HITS analysis assumes that all outlinks on a page are relevant to the given query. In many cases, however, even pages that are among the top K in terms of their similarity to a query Q, may have links to pages that have nothing to do with that particular query. Give a technique that can offset this problem

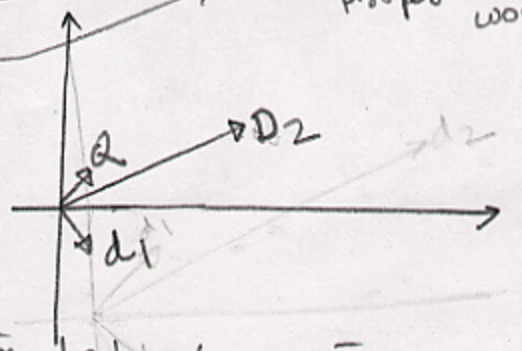
We talked about the idea of weighting the link with the similarity between the Query and the anchor text surrounding the link.



6. [3] Give one good reason why we shouldn't replace cosine-metric with the inverse of eucledian distance (between query and document vectors) for deciding query-document similarity.

Eucledian distance is sensitive to magnitudes of the vectors (rather than to relative proportions of words)

In the example on right Eucledian distance will say Q is closer to d_1 rather than d_2



[If we normalize the vectors before hand, then this argument wont hold – but still inverse of Eucledian distance is not well related to relative proportion of diff words

See the doc-doc distance plot I showed → for B-Regression Example

7. [3] List four (4) magic parameters that Google uses (a magic parameter is a number that needs to be set by Brin & Page—or their underlings).

1) The weight for combining Page rank and similarity with Query α Rank + β Similarity

2) The weights for words appearing in different parts of the Page (anchor, header, ...)

3) The weight for deciding when the surfer follows the link on Page and when surfer does something random.

$$M^* = cM + (1-c)K$$

4) # occurences of a word beyond which the tf weight wont change.