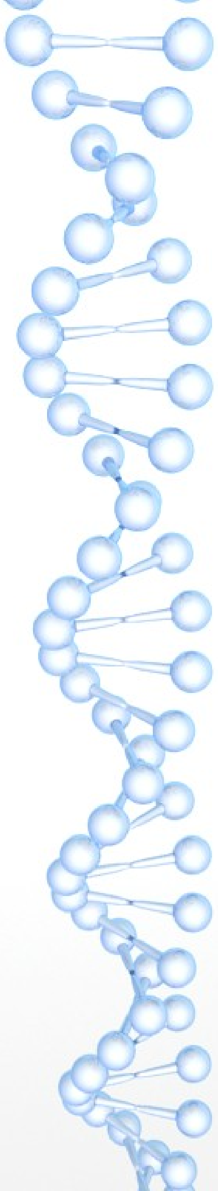
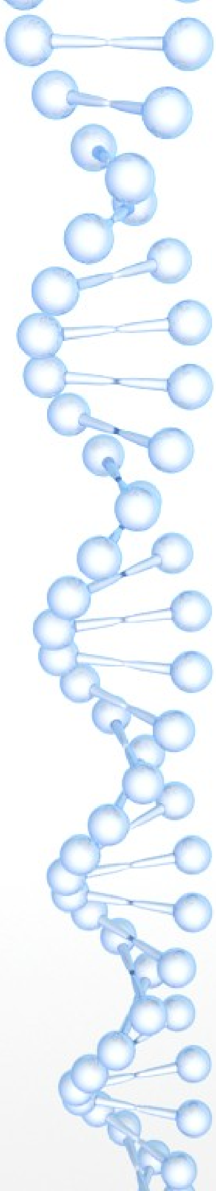


Chapter-10

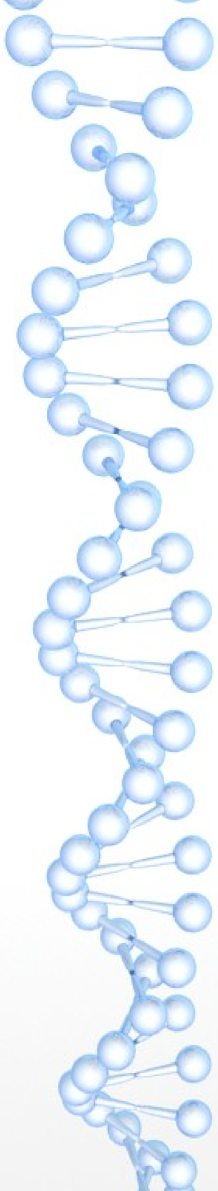
Boolean + Vector Space Query Model
in practice

2021-2022



System-Computed Relevance and Ranking

- Ranking
- The Vector Space Model
- Parameters of Retrieval effectiveness



Ranking

Matching/Ranking of Textual Documents

Major Categories of Methods

1. Exact matching (Boolean)

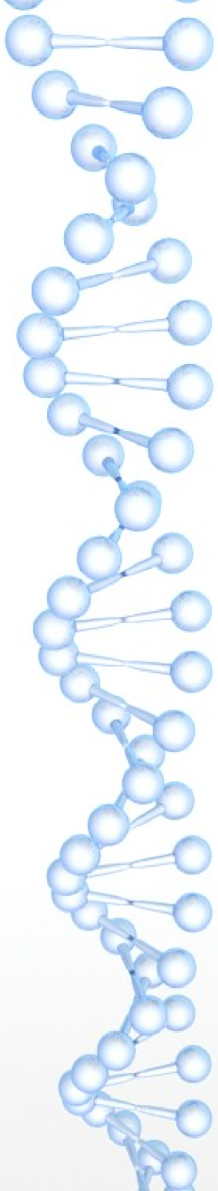
2. Ranking by similarity to query (vector space model)

3. Ranking of matches by importance of documents

(Page Rank)

4. Combination methods

What happens in major search engines (Google rank)



Vector representation of documents and queries

Represents a large space for documents

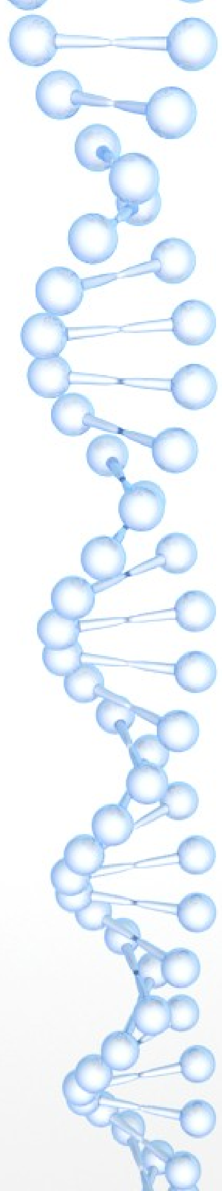
Compare

– Documents

– Documents with queries

Retrieve and rank documents with regards to a specific query

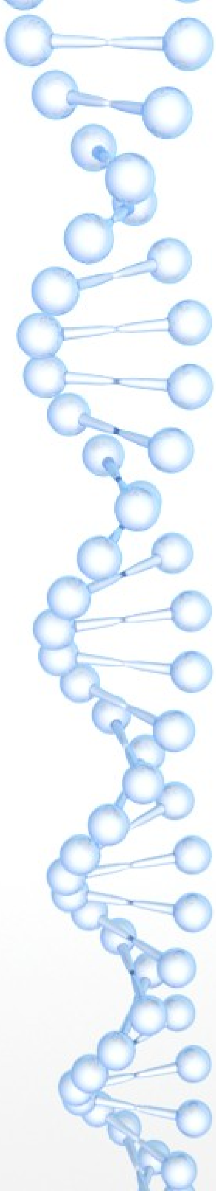
Enables methods of similarity



Vector Model

- Addresses limitations of the boolean model (i.e., binary weights) by assigning non-binary weights to index terms in queries and documents
- Assumption: fixed set of terms used for queries and as document descriptors
- Approach
  - fixed vocabulary consisting of N terms
  - document $D_i = (T_{i1}, T_{i2}, \ldots, T_{ik}, \ldots, T_{iN})$, T_{jk} weight of term k in document i
  - query $Q_j = (Q_{j1}, Q_{j2}, \ldots, Q_{jk}, \ldots, Q_{jN})$, Q_{jk} weight of term k in query j
  - both document and query are interpreted as N-dimensional vectors in the vector space defined by the set of terms
  - similarity of document D_i and query Q_j is defined as the correlation of the two vectors
  - cosine similarity quantifies the correlation using the cosine of the angle between vectors

$$\boxed{sim_{\cos}(D_i, Q_j) = \frac{D_i \bullet Q_j}{|D_i| \times |Q_j|} = \frac{\sum_{k=1}^{N} T_{i,k} \times Q_{j,k}}{\sqrt{\sum_{k=1}^{N} T_{i,k}^2} \times \sqrt{\sum_{k=1}^{N} Q_{j,k}^2}}}$$



Similarity Measures and Relevance

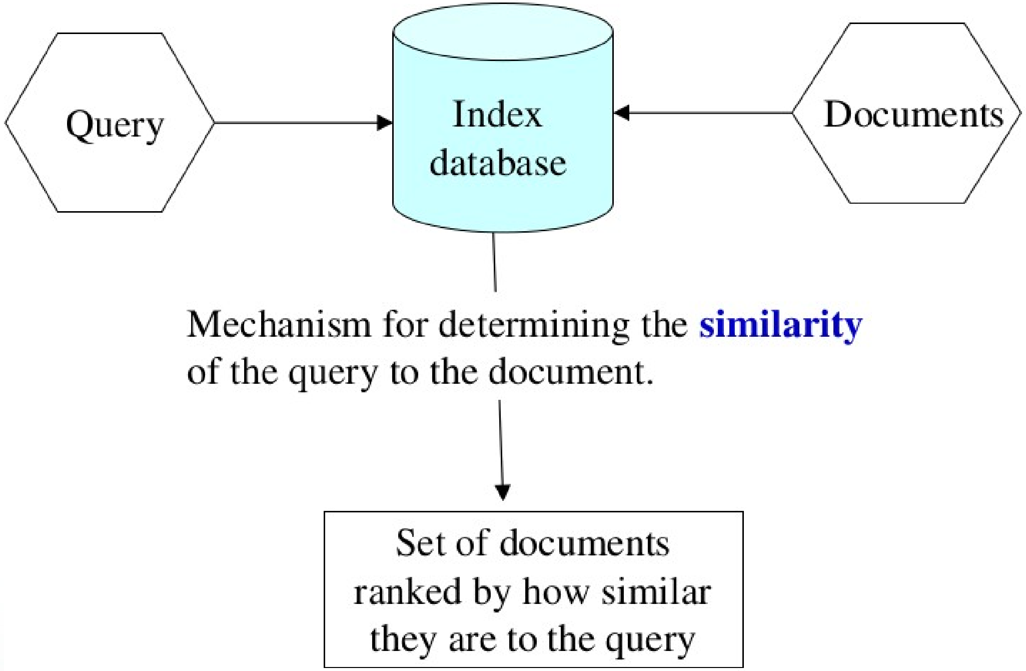
- Retrieve the most similar documents to a query
- Equate similarity to relevance

– Most similar are the most relevant

- This measure is one of “text similarity”

– The matching of text or words

Similarity Ranking Methods



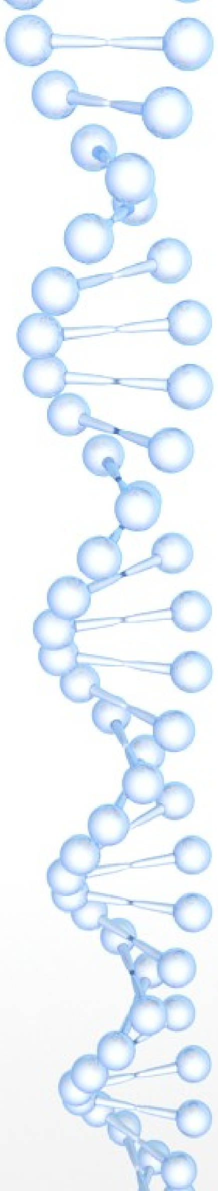
Query

Index database

Documents

Mechanism for determining the **similarity** of the query to the document.

Set of documents ranked by how similar they are to the query



Term Similarity: Example

Problem: Given two text documents, how similar are they?

[Methods that measure similarity do not assume exact matches.]

Example (assume tokens converted to terms)

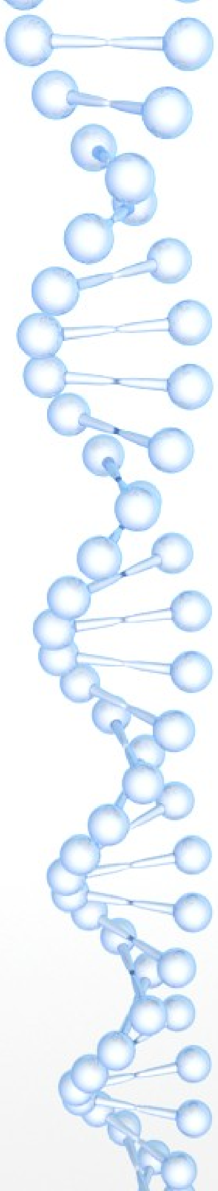
Here are three documents. How similar are they?

D1= {ant ant bee}

D2= {dog bee dog hog dog ant dog}

D3= {cat gnu dog eel fox}

Documents can be any length from one word to thousands.

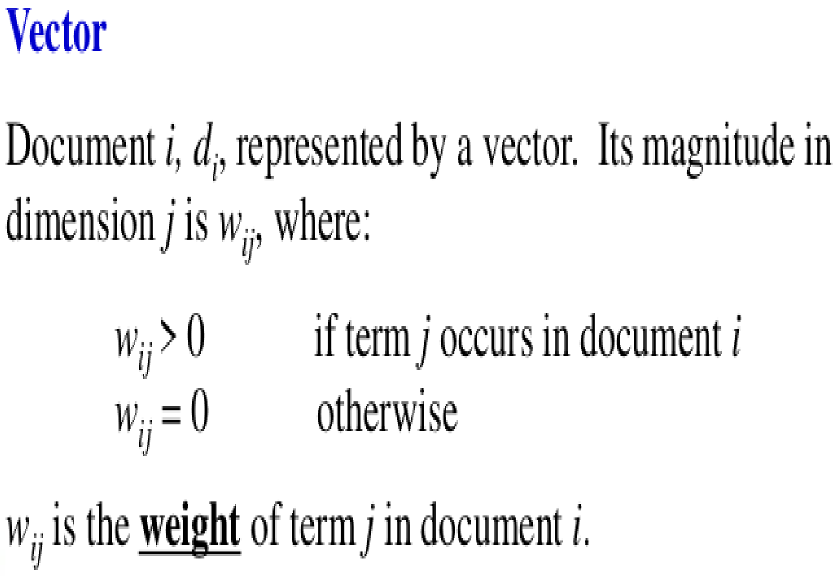


Term Similarity: Basic Concept

- Two documents are similar if they contain some of the same
- terms.
- Possible measures of similarity might take into consideration:
- (a) The lengths of the documents
- (b) The number of terms in common
- (c) Whether the terms are common or unusual
- (d) How many times each term appears

TERM VECTOR SPACE

- **Term vector space**
- n-dimensional space, where n is the number of different terms/tokens used to index a set of documents.



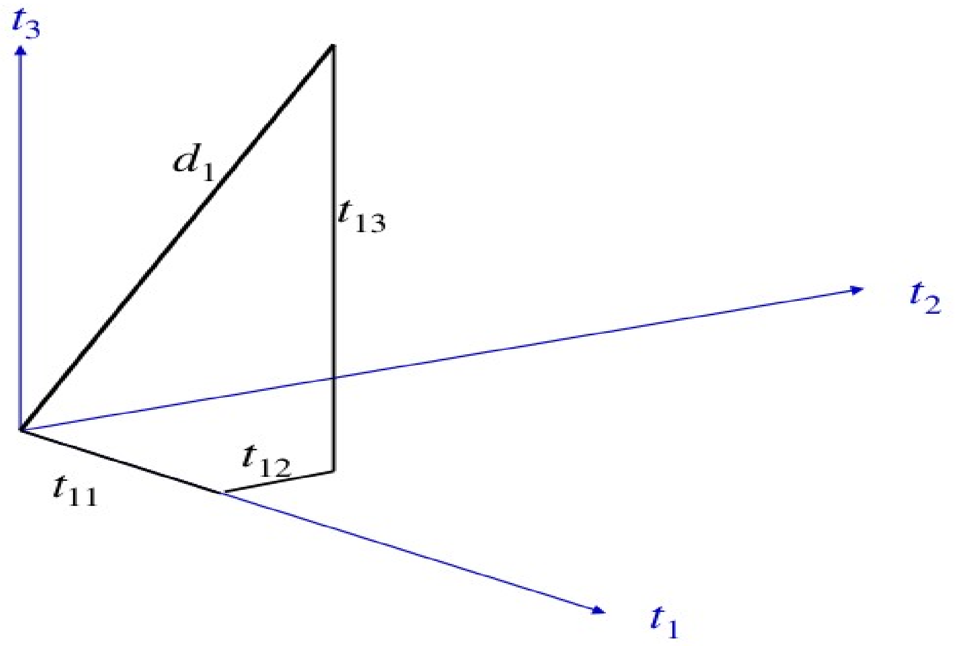
Vector

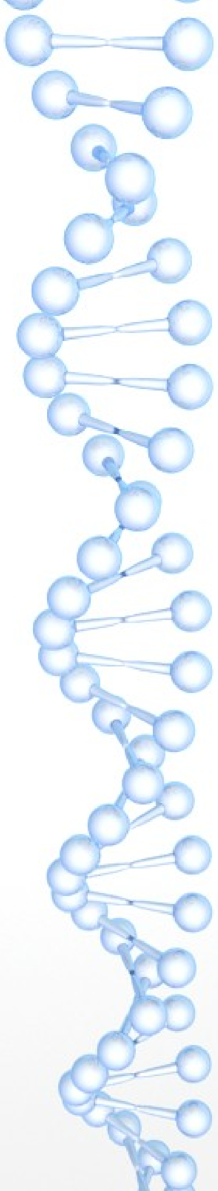
Document *i*, d_i, represented by a vector. Its magnitude in dimension *j* is w_{ij}, where:

$$\begin{array}{ll} w_{ij} > 0 & \text{if term } j \text{ occurs in document } i \\ w_{ij} = 0 & \text{otherwise} \end{array}$$

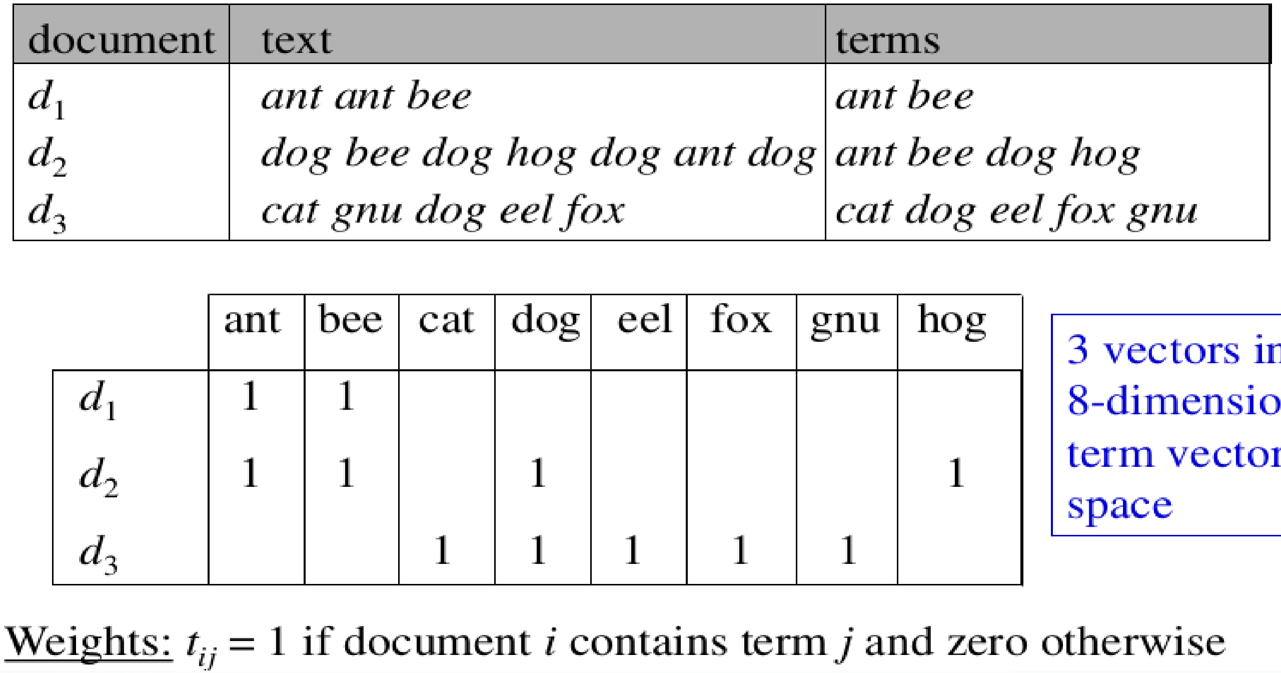
w_{ij} is the **weight** of term *j* in document *i*.

A Document Represented in a 3-Dimensional Term Vector Space





Basic Method: Incidence Matrix (Binary Weighting)



document	text	terms
d_1	*ant ant bee*	*ant bee*
d_2	*dog bee dog hog dog ant dog*	*ant bee dog hog*
d_3	*cat gnu dog eel fox*	*cat dog eel fox gnu*

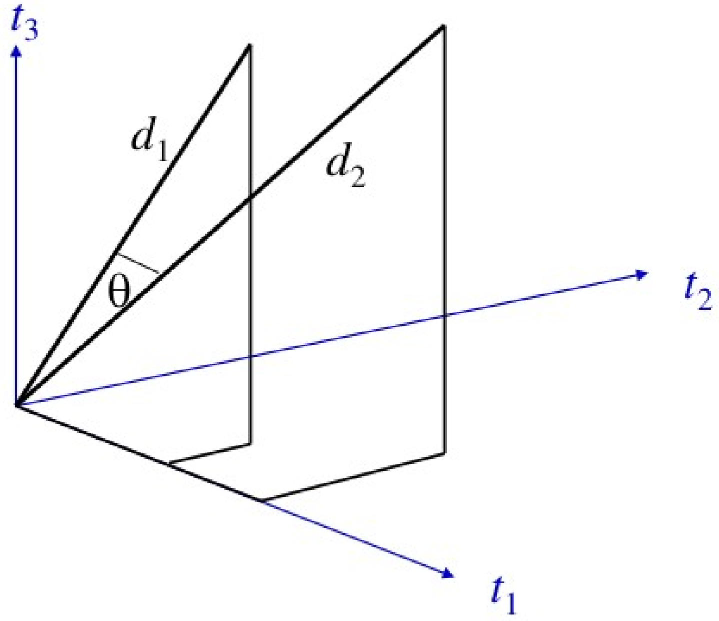
	ant	bee	cat	dog	eel	fox	gnu	hog
d_1	1	1						
d_2	1	1		1				1
d_3			1	1	1	1	1	

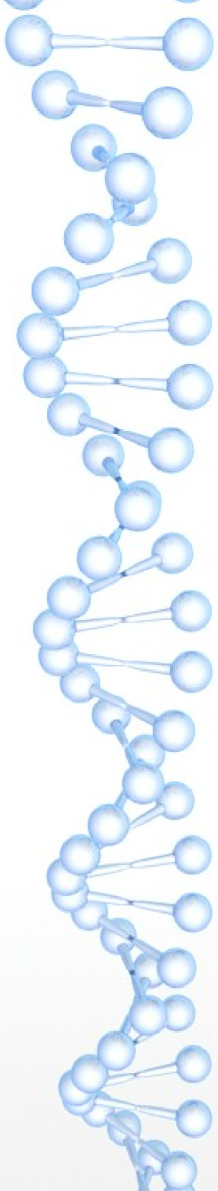
Weights: $t_{ij} = 1$ if document i contains term j and zero otherwise

3 vectors in 8-dimensional term vector space

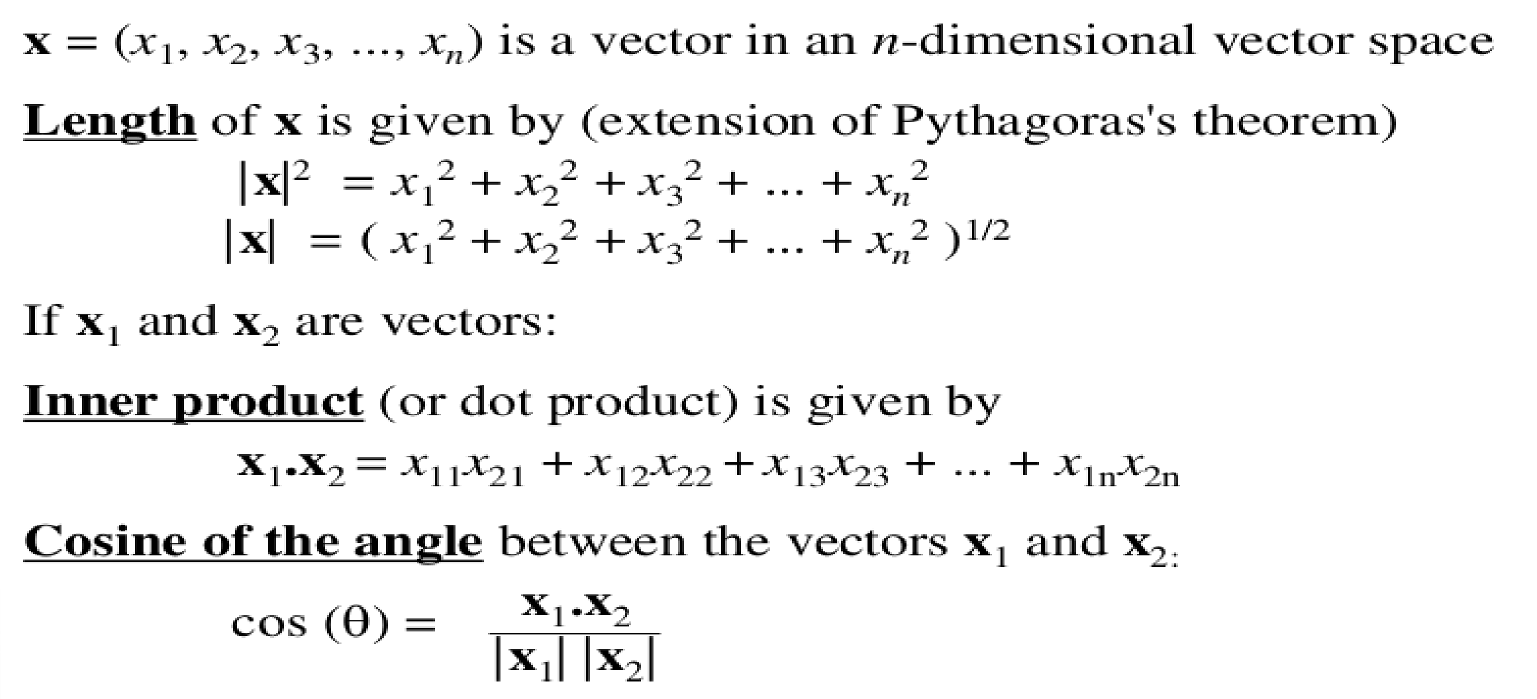
Basic Vector Space Methods: Similarity between 2 documents

The **similarity** between two documents is a function of the **angle** between their **vectors** in the term vector space.





Vector Space Revision



$\mathbf{x} = (x_1, x_2, x_3, \ldots, x_n)$ is a vector in an n-dimensional vector space

Length of **x** is given by (extension of Pythagoras's theorem)

$$|\mathbf{x}|^2 = x_1{}^2 + x_2{}^2 + x_3{}^2 + \ldots + x_n{}^2$$

$$|\mathbf{x}| = (x_1{}^2 + x_2{}^2 + x_3{}^2 + \ldots + x_n{}^2)^{1/2}$$

If $\mathbf{x}_1$ and $\mathbf{x}_2$ are vectors:

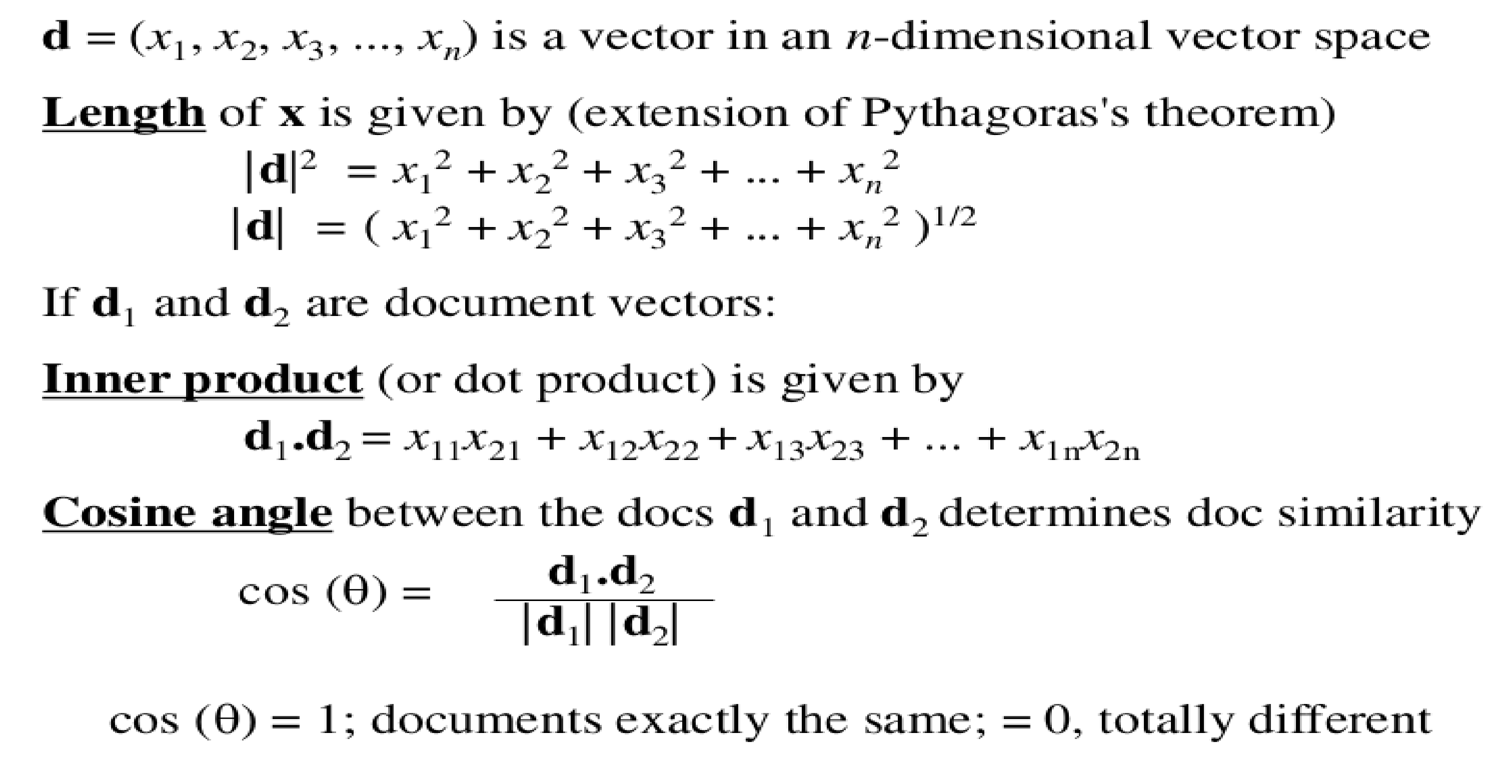
Inner product (or dot product) is given by

$$\mathbf{x}_1.\mathbf{x}_2 = x_{11}x_{21} + x_{12}x_{22} + x_{13}x_{23} + \ldots + x_{1\mathrm{n}}x_{2\mathrm{n}}$$

Cosine of the angle between the vectors $\mathbf{x}_1$ and $\mathbf{x}_{2:}$

$$\cos(\theta) = \frac{\mathbf{x}_1.\mathbf{x}_2}{|\mathbf{x}_1|\,|\mathbf{x}_2|}$$

Document similarity



$\mathbf{d} = (x_1, x_2, x_3, \ldots, x_n)$ is a vector in an n-dimensional vector space

Length of $\mathbf{x}$ is given by (extension of Pythagoras's theorem)

$$|\mathbf{d}|^2 = x_1{}^2 + x_2{}^2 + x_3{}^2 + \ldots + x_n{}^2$$

$$|\mathbf{d}| = (\, x_1{}^2 + x_2{}^2 + x_3{}^2 + \ldots + x_n{}^2\,)^{1/2}$$

If $\mathbf{d}_1$ and $\mathbf{d}_2$ are document vectors:

Inner product (or dot product) is given by

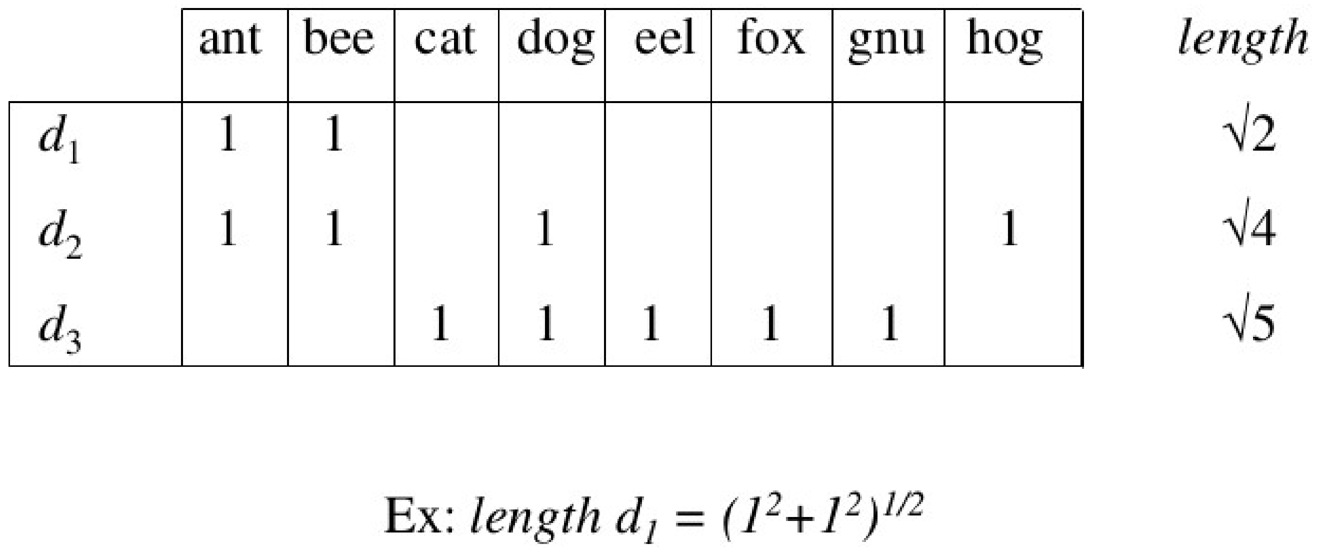
$$\mathbf{d}_1.\mathbf{d}_2 = x_{11}x_{21} + x_{12}x_{22} + x_{13}x_{23} + \ldots + x_{1\mathrm{n}}x_{2\mathrm{n}}$$

Cosine angle between the docs $\mathbf{d}_1$ and $\mathbf{d}_2$ determines doc similarity

$$\cos(\theta) = \frac{\mathbf{d}_1.\mathbf{d}_2}{|\mathbf{d}_1|\,|\mathbf{d}_2|}$$

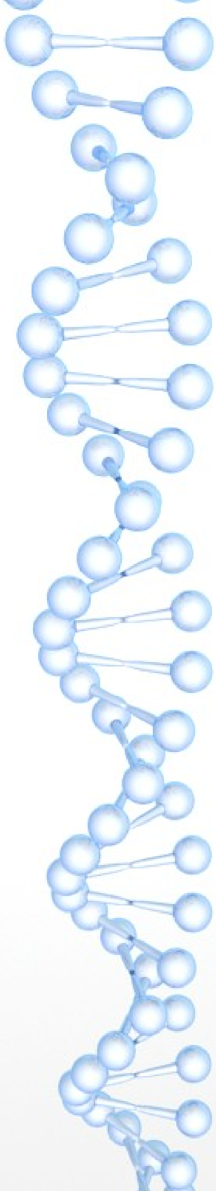
$\cos(\theta) = 1$; documents exactly the same; $= 0$, totally different

Example 1
No Weighting

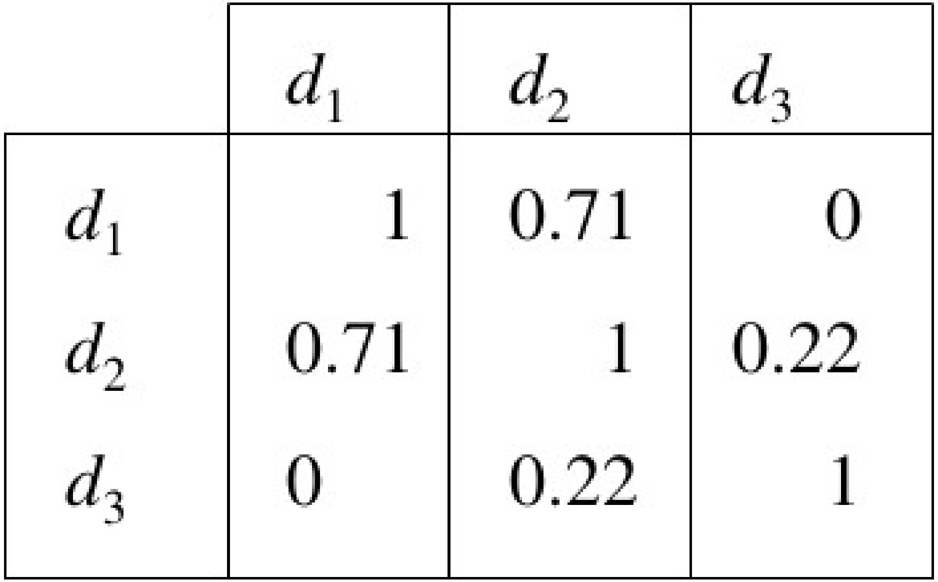


	ant	bee	cat	dog	eel	fox	gnu	hog	*length*
d_1	1	1							√2
d_2	1	1		1				1	√4
d_3			1	1	1	1	1		√5

Ex: *length* $d_1 = (1^2+1^2)^{1/2}$

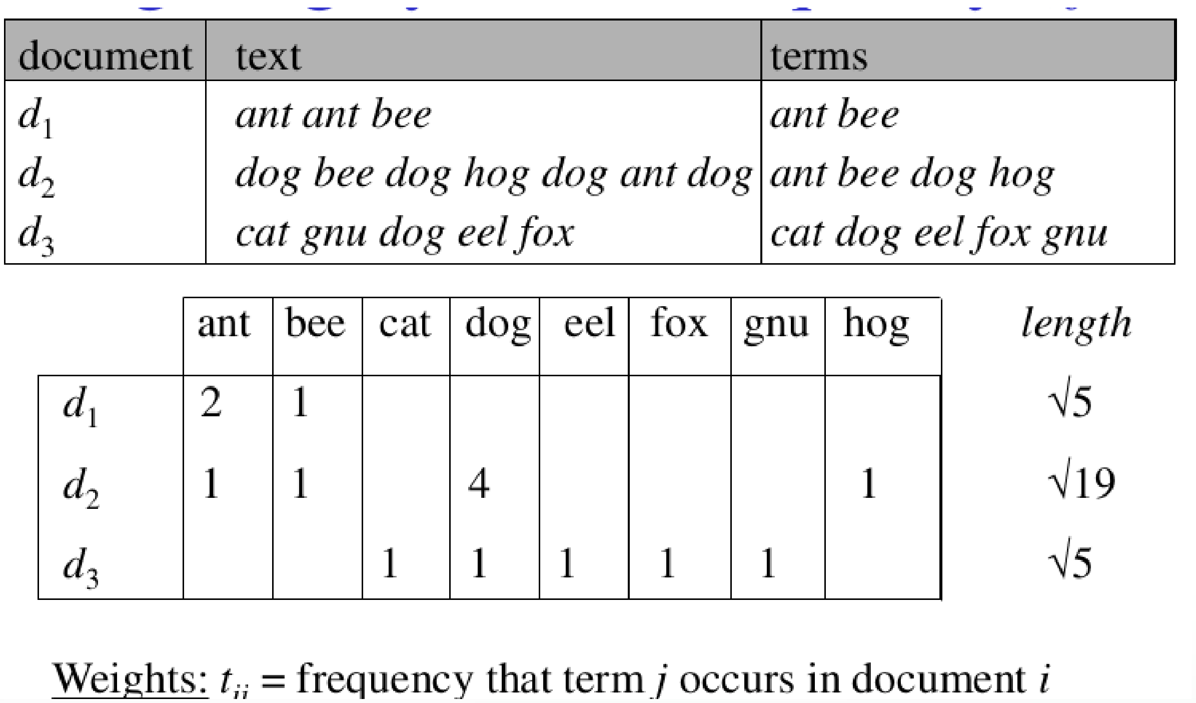


Example-1 (continued)
Similarity of documents in example:



	d_1	d_2	d_3
d_1	1	0.71	0
d_2	0.71	1	0.22
d_3	0	0.22	1

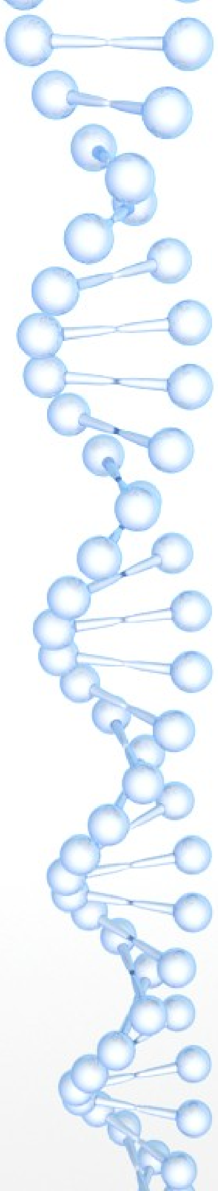
Example 2
Weighting by Term Frequency (tf)



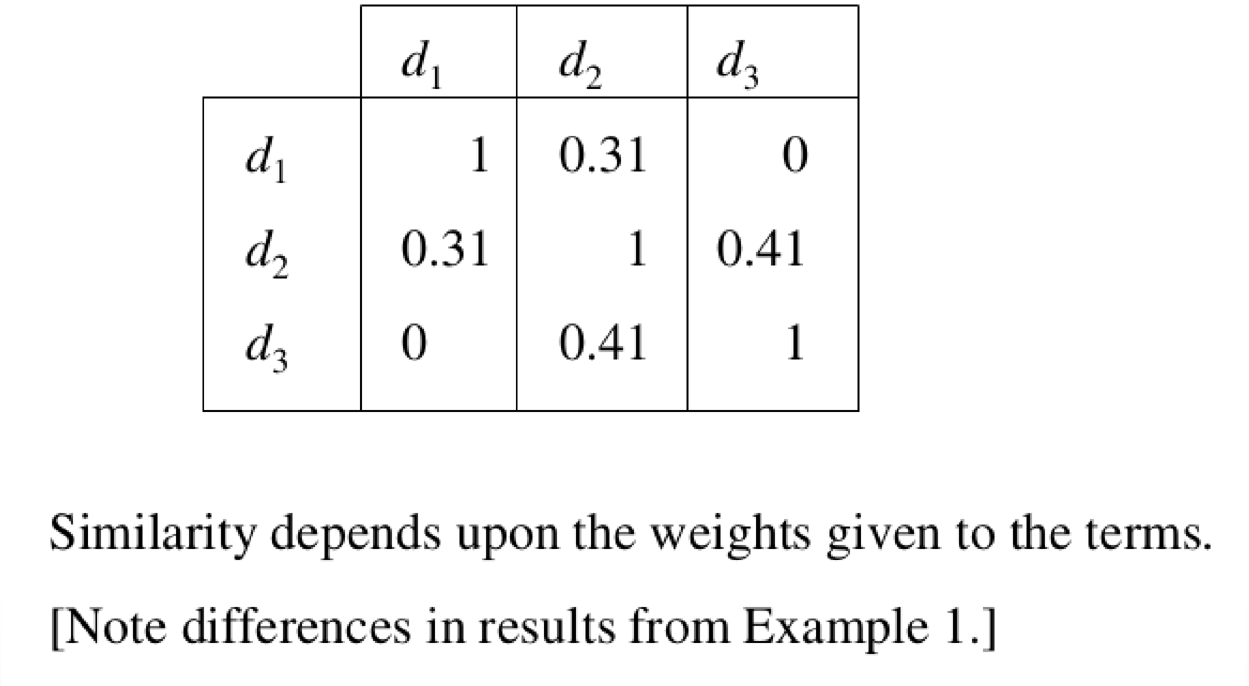
document	text	terms
d_1	*ant ant bee*	*ant bee*
d_2	*dog bee dog hog dog ant dog*	*ant bee dog hog*
d_3	*cat gnu dog eel fox*	*cat dog eel fox gnu*

	ant	bee	cat	dog	eel	fox	gnu	hog	*length*
d_1	2	1							$\sqrt{5}$
d_2	1	1		4				1	$\sqrt{19}$
d_3			1	1	1	1	1		$\sqrt{5}$

Weights: t_{ij} = frequency that term j occurs in document i



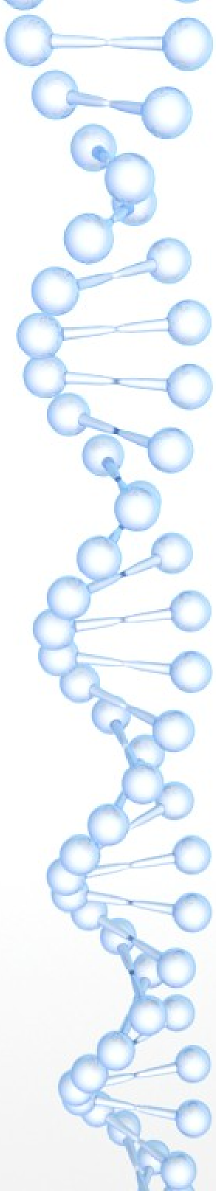
Example 2 (continued)
Similarity of documents in example:



	d_1	d_2	d_3
d_1	1	0.31	0
d_2	0.31	1	0.41
d_3	0	0.41	1

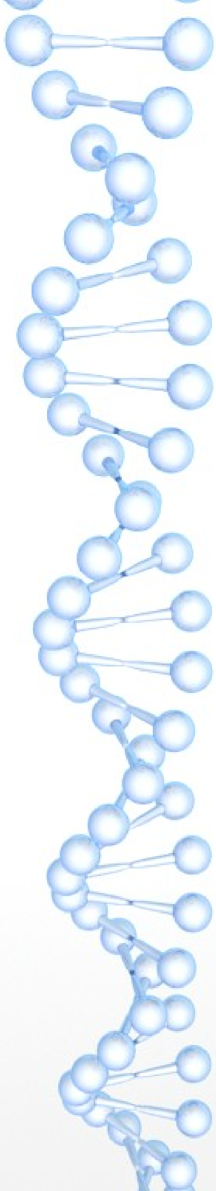
Similarity depends upon the weights given to the terms.

[Note differences in results from Example 1.]

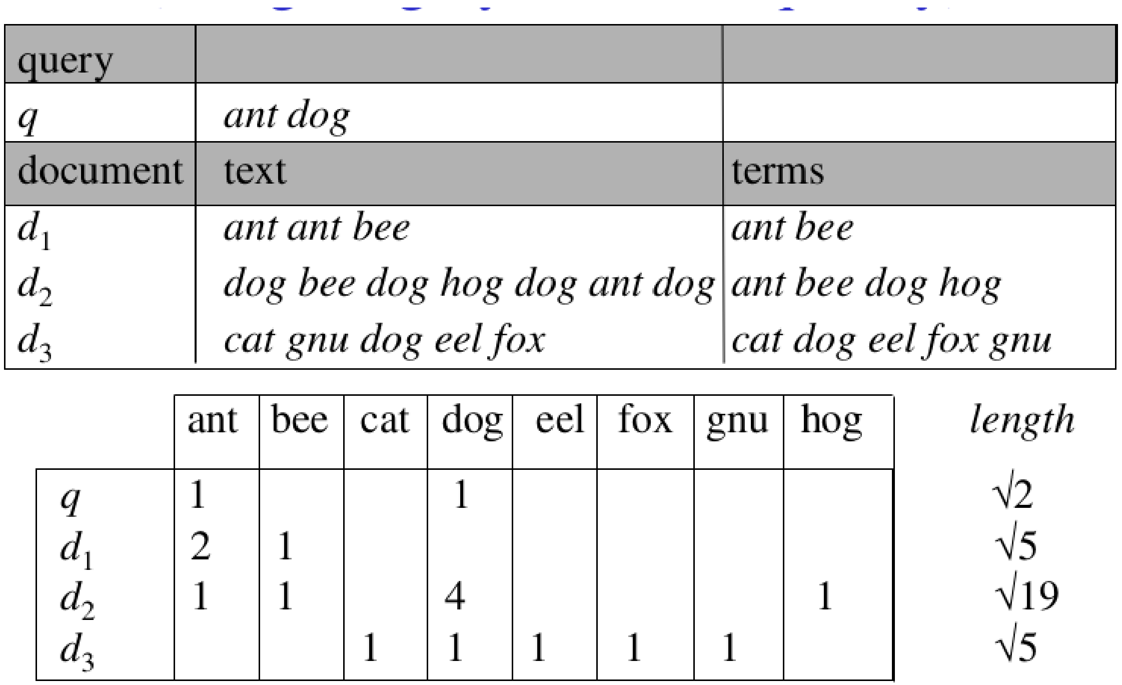


Simple Uses of Vector Similarity in Information Retrieval

- **Threshold**
- For query q, retrieve all documents with similarity
- above a threshold, e.g., similarity > 0.50.
- **Ranking**
- For query q, return the n most similar documents ranked
- in order of similarity.
- [This is the standard practice.]



Simple Example of Ranking (Weighting by Term Frequency)

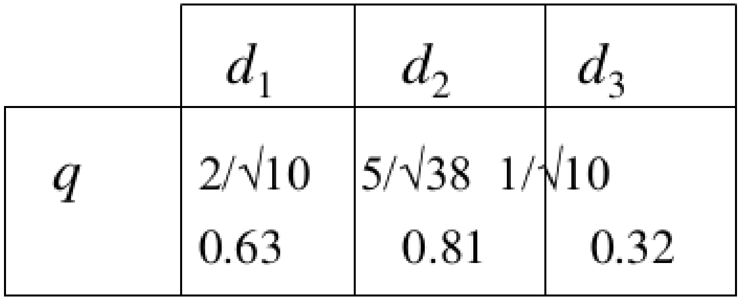


query		
q	*ant dog*	
document	text	terms
d_1	*ant ant bee*	*ant bee*
d_2	*dog bee dog hog dog ant dog*	*ant bee dog hog*
d_3	*cat gnu dog eel fox*	*cat dog eel fox gnu*

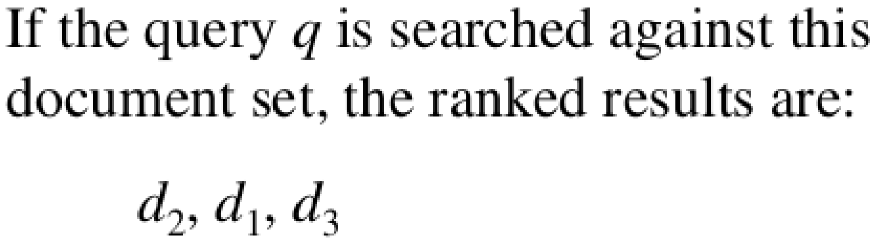
	ant	bee	cat	dog	eel	fox	gnu	hog	*length*
q	1			1					$\sqrt{2}$
d_1	2	1							$\sqrt{5}$
d_2	1	1		4				1	$\sqrt{19}$
d_3			1	1	1	1	1		$\sqrt{5}$

Calculate Ranking

Similarity of query to documents in example:

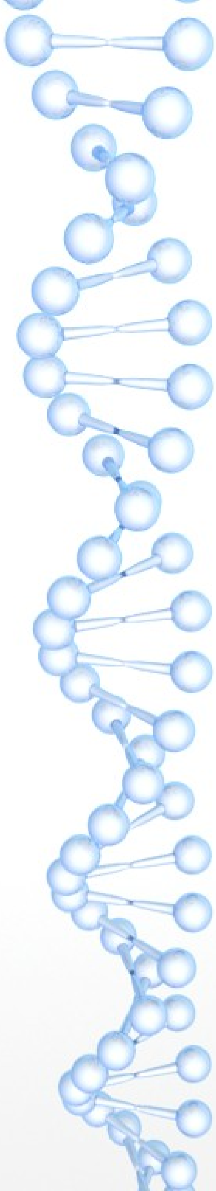


	d_1	d_2	d_3
q	2/√10 0.63	5/√38 0.81	1/√10 0.32

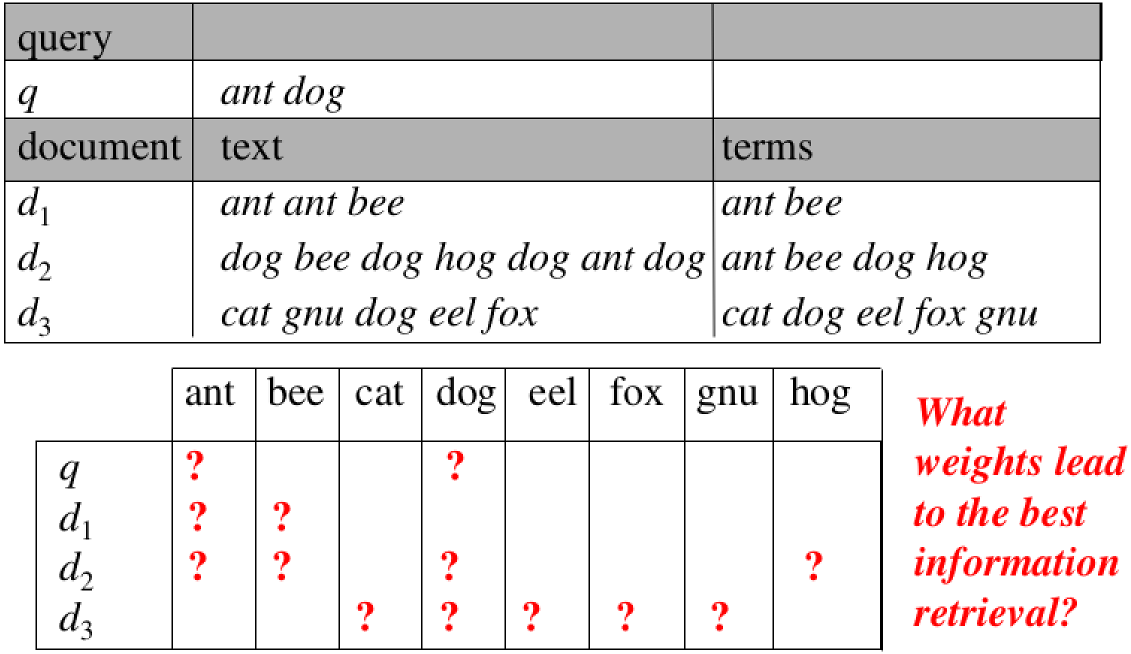


If the query q is searched against this document set, the ranked results are:

d_2, d_1, d_3



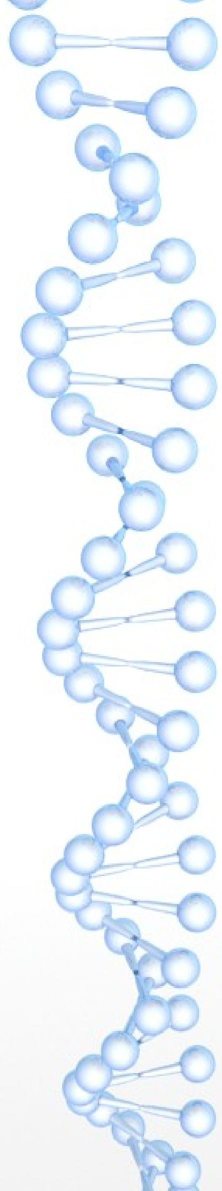
Best Choice of Weights?



query		
q	*ant dog*	
document	**text**	**terms**
d_1	*ant ant bee*	*ant bee*
d_2	*dog bee dog hog dog ant dog*	*ant bee dog hog*
d_3	*cat gnu dog eel fox*	*cat dog eel fox gnu*

	ant	bee	cat	dog	eel	fox	gnu	hog
q	?			?				
d_1	?	?						
d_2	?	?		?				?
d_3			?	?	?	?	?	

What weights lead to the best information retrieval?



Parameters of retrieval effectiveness

- Recall

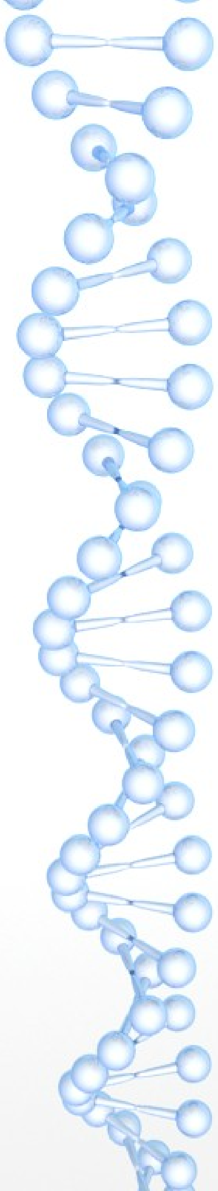
$$R = \frac{\text{Number of relevant items retrieved}}{\text{Total number of relevant items in collection}}$$

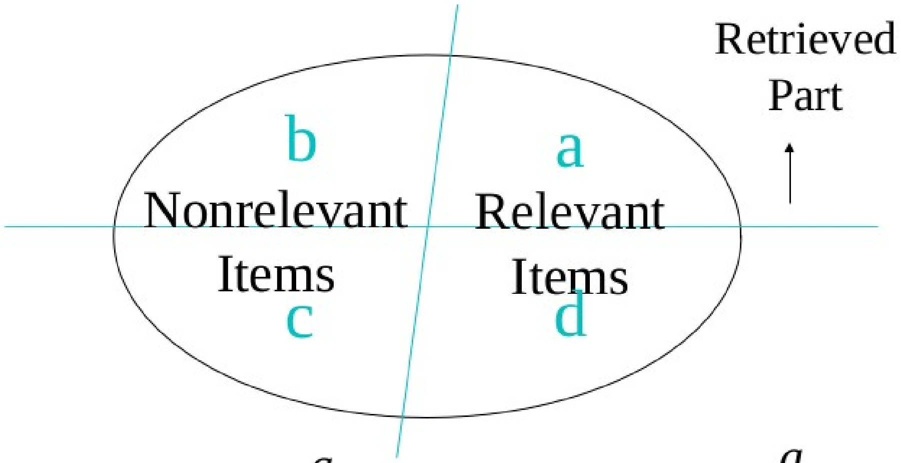
- Precision

$$P = \frac{\text{Number of relevant items retrieved}}{\text{Total number of items retrieved}}$$

- *Goal*

high recall and high precision





Retrieved Part

b

a

Nonrelevant Items

Relevant Items

c

d

$$\text{Recall} = \frac{a}{a+d}$$

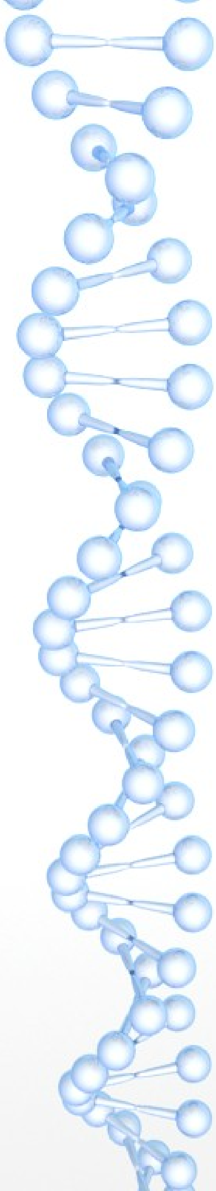
$$\text{Precision} = \frac{a}{a+b}$$

A Joint Measure

- F-score

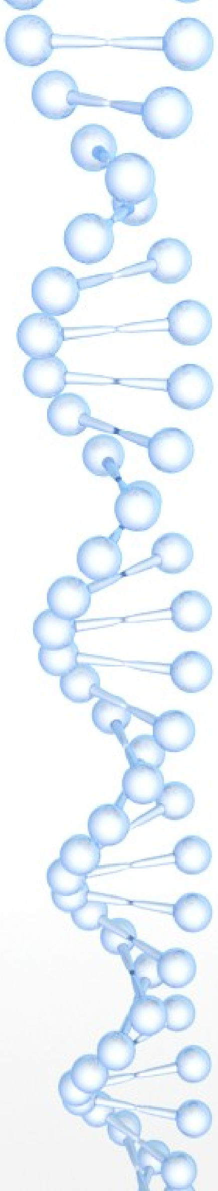
$$F=\frac{(\beta^2+1)\,´P\,´R}{\beta^2\,´P+R}$$

  - β is a parameter that encode the importance of recall and procedure.
  - β=1: equal weight
  - β<1: precision is more important
  - β>1: recall is more important



Choices of Recall and Precision

- Both recall and precision vary from 0 to 1.
- In principle, the average user wants to achieve both
- high recall and high precision.
- In practice, a compromise must be reached because
- simultaneously optimizing recall and precision is not
- normally achievable.



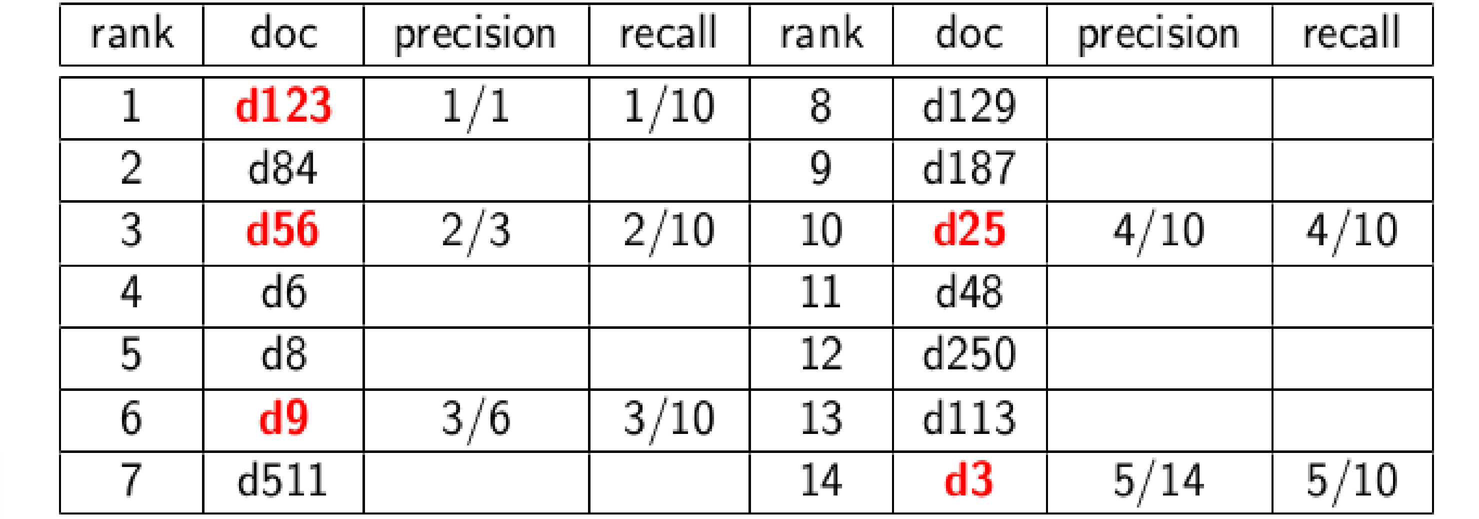
Recall / Precision-Example

- Let us assume that for a given query, the following documents are relevant (10 relevant documents):

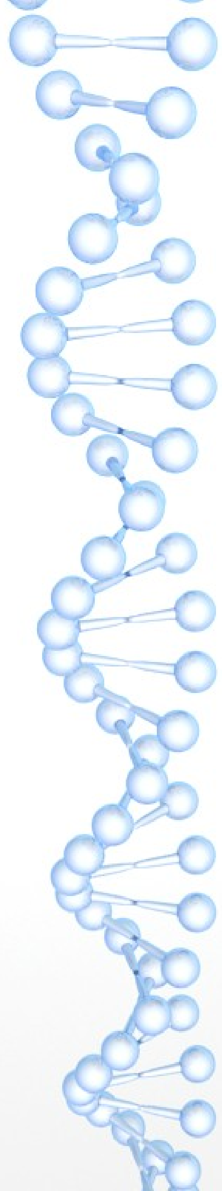
  {d3, d5, d9, d25, d39, d44, d56, d71, d89, d123}

{d3, d5, d9, d25, d39, d44, d56, d71, d89, d123}

- Now suppose that the following documents are retrieved for that query:



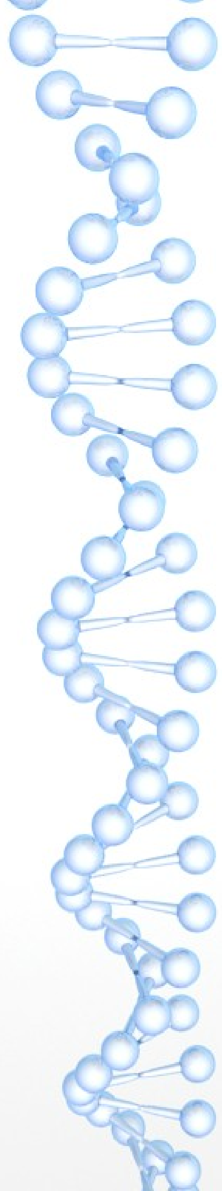
rank	doc	precision	recall	rank	doc	precision	recall
1	**d123**	1/1	1/10	8	d129		
2	d84			9	d187		
3	**d56**	2/3	2/10	10	**d25**	4/10	4/10
4	d6			11	d48		
5	d8			12	d250		
6	**d9**	3/6	3/10	13	d113		
7	d511			14	**d3**	5/14	5/10



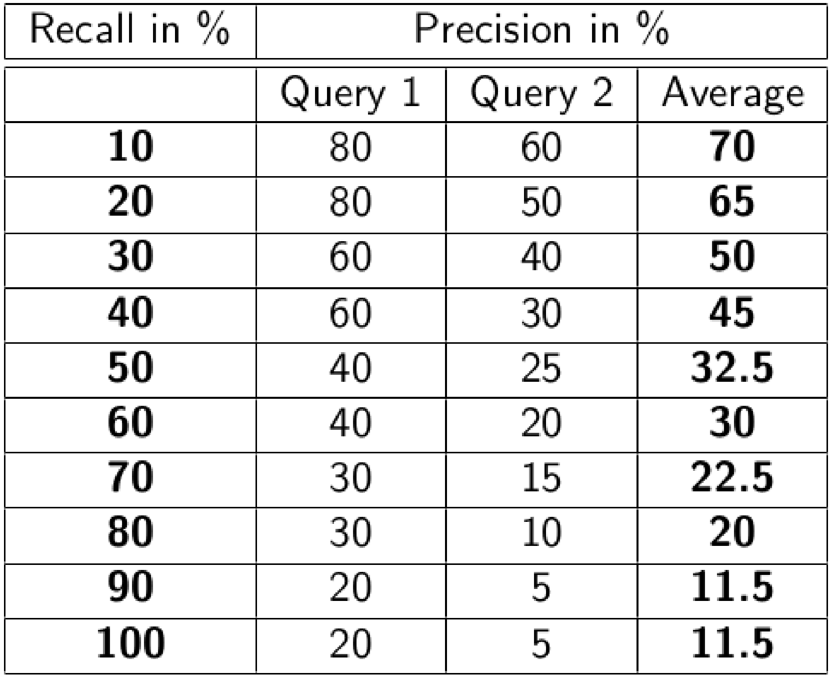
Example Explaining

- For each relevant document (in red bold), we calculate the precision value and the recall value.
- For example, for d56, we have 3 retrieved documents, and 2 among them are relevant, so the precision is 2/3.
- We have 2 of the relevant documents so far retrieved (the total number of relevant documents being 10), so recall is 2/10.

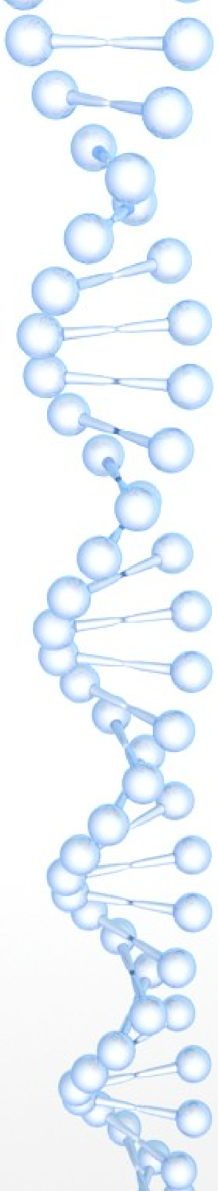
- For each query, we obtain pairs of recall and precision values
- In our example, we would obtain (1/10, 1/1) (2/10, 2/3) (3/10, 3/6) (4/10, 4/10) (5/10, 5/14) . . . which are usually expressed in % (10%, 100%) (20%, 66.66%) (30%, 50%) (40%, 40%) (50%, 35.71%) . . .
- This can be read for instance: at 20% recall, we have 66.66% precision; at 50% recall, we have 35.71% precision



Averaging



Recall in %	Precision in %		
	Query 1	Query 2	Average
10	80	60	**70**
20	80	50	**65**
30	60	40	**50**
40	60	30	**45**
50	40	25	**32.5**
60	40	20	**30**
70	30	15	**22.5**
80	30	10	**20**
90	20	5	**11.5**
100	20	5	**11.5**



End of Chapter-10

Any Question….?