CIS 833 – Information Retrieval and Text Mining

Lecture 10

Latent Semantic Indexing

September 24, 2015

Credits for slides: Hofmann, Mihalcea, Mobasher, Mooney, Schutze.

Assignments

- HW2 due September 25th
- PA1 due October 7th
- Exam 1 – October 13th

Classes of Retrieval Models

- Boolean models (set theoretic)
  - Extended Boolean
- Vector space models (algebraic)
  - Generalized VS
  - Latent Semantic Indexing
- Probabilistic models
  - Inference Networks
  - Belief Networks

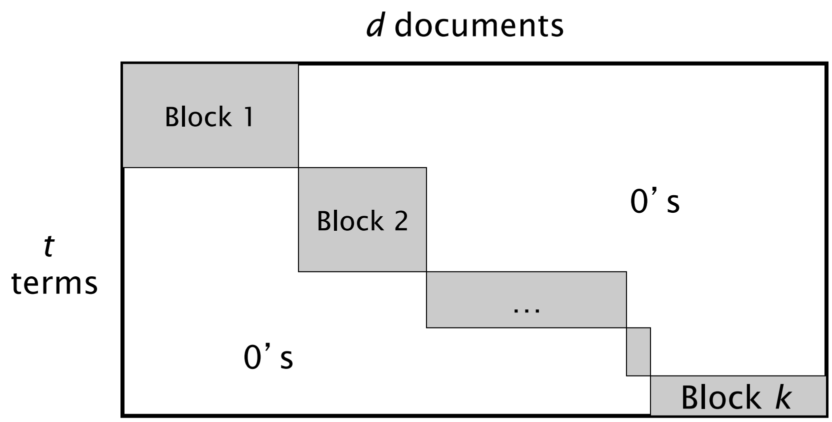
Exact match

Ranking -
"Best" match

Required Reading

- Textbook - Chapter 18 (latent semantic indexing)
- Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman, "Indexing by latent semantic analysis". *Journal of the American Society for Information Science, Volume 41, Issue 6, 1990*

Intuition from block matrices



Vocabulary partitioned into *k* topics (clusters); each doc discusses only one topic.

Latent Semantic Indexing

Variant of the vector space model

Objective

Replace indexes that use **sets of terms** by indexes that use **concepts**

Approach

Map the term vector space into a lower dimensional space, using singular value decomposition.

https://en.wikipedia.org/wiki/Singular_value_decomposition

Each dimension in the new space corresponds to a latent concept in the original data - uncorrelated, significant basis vectors

Replace original words with a subset of the new concepts (say 100, but the number may vary) in both documents and queries

Compute similarities in this new space

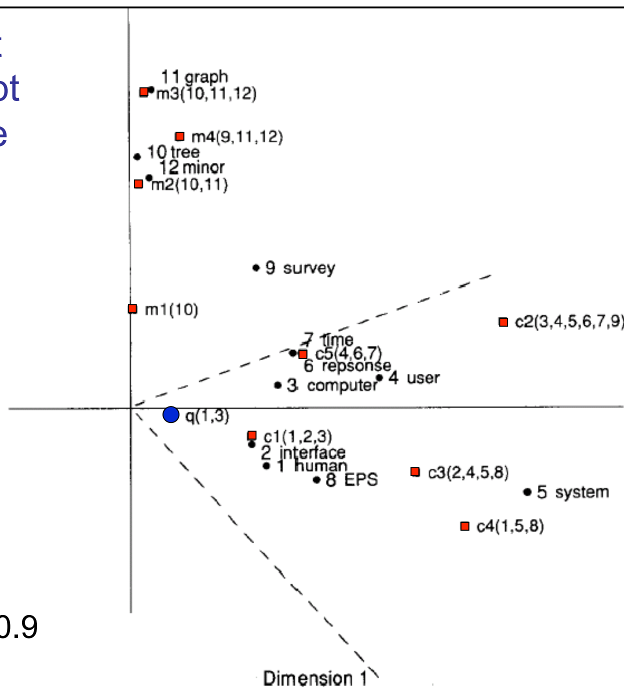
Computationally expensive, uncertain effectiveness

[Deerwester et al., 1990]

Technical Memo Example: Titles

c1	*Human* machine *interface* for Lab ABC *computer* applications
c2	A *survey* of *user* opinion of *computer system response time*
c3	The *EPS user interface* management *system*
c4	*System* and *human system* engineering testing of *EPS*
c5	Relation of *user*-perceived *response time* to error measurement
m1	The generation of random, binary, unordered *trees*
m2	The intersection *graph* of paths in *trees*
m3	*Graph minors* IV: Widths of *trees* and well-quasi-ordering
m4	*Graph minors*: A *survey*

Latent Concept Space



- term
- document
- query
- --- cosine > 0.9

Mathematical concepts

Define $\boldsymbol{X}$ as the term-document matrix, with t rows (number of index terms) and d columns (number of documents).

Singular Value Decomposition

For any matrix $\boldsymbol{X}$, with t rows and d columns, there exist matrices $\boldsymbol{T}_0$, $\boldsymbol{S}_0$ and $\boldsymbol{D}_0$, such that:

$$\boldsymbol{X} = \boldsymbol{T}_0 \boldsymbol{S}_0 \boldsymbol{D}_0'$$

$\boldsymbol{T}_0$ and $\boldsymbol{D}_0$ are the matrices of left and right <u>singular vectors</u>

$\boldsymbol{T}_0$ and $\boldsymbol{D}_0$ have orthogonal, unit-length columns:

$\boldsymbol{T}_0' \boldsymbol{T}_0 = \mathrm{I}$ and $\boldsymbol{D}_0' \boldsymbol{D}_0 = \mathrm{I}$

$\boldsymbol{S}_0$ is the diagonal matrix of <u>singular values</u>

LSI: example

$$T_0 = \begin{matrix}
0.22 & -0.11 & 0.29 & -0.41 & -0.11 & -0.34 & 0.52 & -0.06 & -0.41 \\
0.20 & -0.07 & 0.14 & -0.55 & 0.28 & 0.50 & -0.07 & -0.01 & -0.11 \\
0.24 & 0.04 & -0.16 & -0.59 & -0.11 & -0.25 & -0.30 & 0.06 & 0.49 \\
0.40 & 0.06 & -0.34 & 0.10 & 0.33 & 0.38 & 0.00 & 0.00 & 0.01 \\
0.64 & -0.17 & 0.36 & 0.33 & -0.16 & -0.21 & -0.17 & 0.03 & 0.27 \\
0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\
0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\
0.30 & -0.14 & 0.33 & 0.19 & 0.11 & 0.27 & 0.03 & -0.02 & -0.17 \\
0.21 & 0.27 & -0.18 & -0.03 & -0.54 & 0.08 & -0.47 & -0.04 & -0.58 \\
0.01 & 0.49 & 0.23 & 0.03 & 0.59 & -0.39 & -0.29 & 0.25 & -0.23 \\
0.04 & 0.62 & 0.22 & 0.00 & -0.07 & 0.11 & 0.16 & -0.68 & 0.23 \\
0.03 & 0.45 & 0.14 & -0.01 & -0.30 & 0.28 & 0.34 & 0.68 & 0.18
\end{matrix}$$

$$S_0 = \begin{matrix}
3.34 & & & & & & & & \\
& 2.54 & & & & & & & \\
& & 2.35 & & & & & & \\
& & & 1.64 & & & & & \\
& & & & 1.50 & & & & \\
& & & & & 1.31 & & & \\
& & & & & & 0.85 & & \\
& & & & & & & 0.56 & \\
& & & & & & & & 0.36
\end{matrix}$$

$$D_0 = \begin{matrix}
0.20 & -0.06 & 0.11 & -0.95 & 0.05 & -0.08 & 0.18 & -0.01 & -0.06 \\
0.61 & 0.17 & -0.50 & -0.03 & -0.21 & -0.26 & -0.43 & 0.05 & 0.24 \\
0.46 & -0.03 & 0.21 & 0.04 & 0.38 & 0.72 & -0.24 & 0.01 & 0.02 \\
0.54 & -0.23 & 0.57 & 0.27 & -0.21 & -0.37 & 0.26 & -0.02 & -0.08 \\
0.28 & 0.11 & -0.51 & 0.15 & 0.33 & 0.03 & 0.67 & -0.06 & -0.26 \\
0.00 & 0.19 & 0.10 & 0.02 & 0.39 & -0.30 & -0.34 & 0.45 & -0.62 \\
0.01 & 0.44 & 0.19 & 0.02 & 0.35 & -0.21 & -0.15 & -0.76 & 0.02 \\
0.02 & 0.62 & 0.25 & 0.01 & 0.15 & 0.00 & 0.25 & 0.45 & 0.52 \\
0.08 & 0.53 & 0.08 & -0.03 & -0.60 & 0.36 & -0.04 & -0.07 & -0.45
\end{matrix}$$

Dimensions of matrices

$$\underset{t \times d}{\boldsymbol{X}} = \underset{t \times m}{\boldsymbol{T}_0} \; \underset{m \times m}{\boldsymbol{S}_0} \; \underset{m \times d}{\boldsymbol{D}_0'}$$

m is the rank of $\boldsymbol{X} \leqslant \min(t, d)$

Reduced Rank

$\boldsymbol{S}_0$ can be chosen so that the diagonal elements are positive and decreasing in magnitude. Keep the first k and set the others to zero.

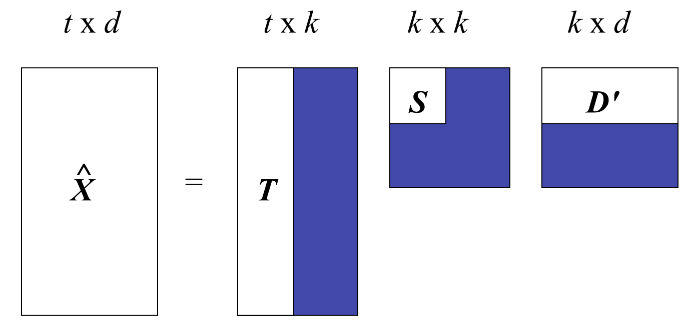
Delete the zero rows and columns of $\boldsymbol{S}_0$ and the corresponding rows and columns of $\boldsymbol{T}_0$ and $\boldsymbol{D}_0$. This gives:

$$\boldsymbol{X} \approx \hat{\boldsymbol{X}} = \boldsymbol{TSD}'$$

Interpretation

If value of k is selected well, expectation is that $\hat{\boldsymbol{X}}$ retains the semantic information from $\boldsymbol{X}$, but eliminates noise from synonymy and recognizes dependence.

Dimensionality Reduction: Selection of singular values



k is the number of latent concepts (singular values) chosen to represent the document (typically 300 ~ 500)

Usually, k « m

$X \sim \hat{X} = TSD'$ - an individual cell of X is the "number of occurrences" of term i in document j.

LSI: example

$X \simeq$

$$T = \begin{bmatrix} 0.22 & -0.11 \\ 0.20 & -0.07 \\ 0.24 & 0.04 \\ 0.40 & 0.06 \\ 0.64 & -0.17 \\ 0.27 & 0.11 \\ 0.27 & 0.11 \\ 0.30 & -0.14 \\ 0.21 & 0.27 \\ 0.01 & 0.49 \\ 0.04 & 0.62 \\ 0.03 & 0.45 \end{bmatrix} \quad S = \begin{bmatrix} 3.34 & \\ & 2.54 \end{bmatrix} \quad D' = \begin{bmatrix} 0.20 & 0.61 & 0.46 & 0.54 & 0.28 & 0.00 & 0.02 & 0.02 & 0.08 \\ -0.06 & 0.17 & -0.13 & -0.23 & 0.11 & 0.19 & 0.44 & 0.62 & 0.53 \end{bmatrix}$$

$$\hat{X} = \begin{bmatrix} 0.16 & 0.40 & 0.38 & 0.47 & 0.18 & -0.05 & -0.12 & -0.16 & -0.09 \\ 0.14 & 0.37 & 0.33 & 0.40 & 0.16 & -0.03 & -0.07 & -0.10 & -0.04 \\ 0.15 & 0.51 & 0.36 & 0.41 & 0.24 & 0.02 & 0.06 & 0.09 & 0.12 \\ 0.26 & 0.84 & 0.61 & 0.70 & 0.39 & 0.03 & 0.08 & 0.12 & 0.19 \\ 0.45 & 1.23 & 1.05 & 1.27 & 0.56 & -0.07 & -0.15 & -0.21 & -0.05 \\ 0.16 & 0.58 & 0.38 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\ 0.16 & 0.58 & 0.38 & 0.42 & 0.28 & 0.06 & 0.13 & 0.19 & 0.22 \\ 0.22 & 0.55 & 0.51 & 0.63 & 0.24 & -0.07 & -0.14 & -0.20 & -0.11 \\ 0.10 & 0.53 & 0.23 & 0.21 & 0.27 & 0.14 & 0.31 & 0.44 & 0.42 \\ -0.06 & 0.23 & -0.14 & -0.27 & 0.14 & 0.24 & 0.55 & 0.77 & 0.66 \\ -0.06 & 0.34 & -0.15 & -0.30 & 0.20 & 0.31 & 0.69 & 0.98 & 0.85 \\ -0.04 & 0.25 & -0.10 & -0.21 & 0.15 & 0.22 & 0.50 & 0.71 & 0.62 \end{bmatrix}$$

Comparing original and LSI

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

human	0.16	0.40	0.38	0.47	0.18	−0.05	−0.12	−0.16	−0.09
interface	0.14	0.37	0.33	0.40	0.16	−0.03	−0.07	−0.10	−0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	−0.07	−0.15	−0.21	−0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	−0.07	−0.14	−0.20	−0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	−0.06	0.23	−0.14	−0.27	0.14	0.24	0.55	0.77	0.66
graph	−0.06	0.34	−0.15	−0.30	0.20	0.31	0.69	0.98	0.85
minors	−0.04	0.25	−0.10	−0.21	0.15	0.22	0.50	0.71	0.62

Calculating Similarities in the Concept Space

Objective:

Calculate similarities between documents and queries, using the matrices **T**, **S**, and **D**.

Calculating Similarities in the Concept Space

Calculate:

- similarity between two terms (e.g., to form a concept hierarchy)

- similarity between two documents (e.g., to cluster documents into groups)

- similarity between a query and a document (e.g., in information retrieval)

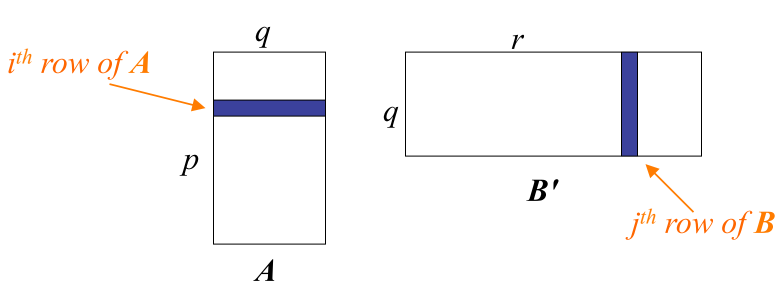
using matrices **T**, **S**, and **D**.

Mathematical Fact

$\boldsymbol{A}$ is a p x q matrix
$\boldsymbol{B}$ is a r x q matrix

$\mathbf{a}_i$ is the vector represented by row i of $\boldsymbol{A}$
$\mathbf{b}_j$ is the vector represented by row j of $\boldsymbol{B}$

The inner product $\mathbf{a}_i \cdot \mathbf{b}_j$ is element i, j of $\boldsymbol{AB'}$



Comparing Two Terms

The dot product of two rows of $\hat{X}$ reflects the extent to which two terms have a similar pattern of occurrences.

$\hat{X}\hat{X}' = TSD'(TSD')'$

$= TSD'DS'T'$

$= TSS'T'$ Since D is orthonormal

$= TS(TS)'$

To calculate the i, j cell, take the dot product between the i and j rows of TS

Since S is diagonal, TS differs from T only by stretching the coordinate system

Comparing Two Documents

The dot product of two columns of $\hat{X}$ reflects the extent to which two columns have a similar pattern of occurrences.

$\hat{X}'\hat{X} = (TSD')'TSD'$

$= DS(DS)'$

To calculate the i, j cell, take the dot product between the i and j columns of DS.

Since S is diagonal DS differs from D only by stretching the coordinate system

Comparing a Query and a Document

A **query** can be expressed as a vector $\boldsymbol{x}_q$ in the **term-document vector space.**

x_{qi} = 1 if term i is in the query and 0 otherwise.

(Ignore query terms that are not in the term vector space.)

Let p_{qj} be the **inner product** of the **query** $\boldsymbol{x}_q$ with **document** $\boldsymbol{d}_j$ in the term-document vector space.

p_{qj} is the j^th element in the product of $\boldsymbol{x}_q'\hat{\boldsymbol{X}}$.

Comparing a Query and a Document

$$[p_{q1} \cdots p_{qj} \cdots p_{qd}] = [x_{q1}\ x_{q2} \cdots x_{qt}] \left[\quad \hat{X} \quad \right]$$

inner product of query q with document d_j

query

document d_j is column j of $\hat{X}$

$$\begin{aligned} \boldsymbol{p}_q' &= \boldsymbol{x}_q'\hat{\boldsymbol{X}} \\ &= \boldsymbol{x}_q'\boldsymbol{TSD}' \\ &= \boldsymbol{x}_q'\boldsymbol{T}(\boldsymbol{DS})' \end{aligned}$$

$$similarity(q, d_j) = \frac{p_{qj}}{|\boldsymbol{x}_q|\ |\boldsymbol{d}_j|}$$

cosine of angle is inner product divided by lengths of vectors

Comparing a Query and a Document

Alternatively, treat the query q as a **pseudo-document d_q** in the **concept space**:

$$d_q = x_q' T S^{-1}$$

$$d_{q(1\text{x}k)} = x_q'{}_{(1\text{x}t)}\, T_{(t\text{x}k)}\, S^{-1}{}_{(k\text{x}k)}$$

To compare a query against document j, extend the method used to compare document i with document j.

Take the j^{th} element of the product of:

$$d_q S \text{ and } (DS)'$$

This is the j^{th} element of product of:

$x_q' T\,(DS)'$ which is the same expression as before.

Technical Memo Example: Query

Terms	Query $\mathbf{x}_q$
human	1
interface	0
computer	0
user	0
system	1
response	0
time	0
EPS	0
survey	0
trees	1
graph	0
minors	0

Query:
"human system interactions on trees"

In **term-document** space, a query is represented by $\mathbf{x}_q$, a column vector with t elements.

In **concept space**, a query is represented by $\mathbf{d}_q$, a row vector with k elements.

Experimental Results

Deerwester, et al. tried latent semantic indexing on two test collections, MED and CISI, where queries and relevant judgments were available.

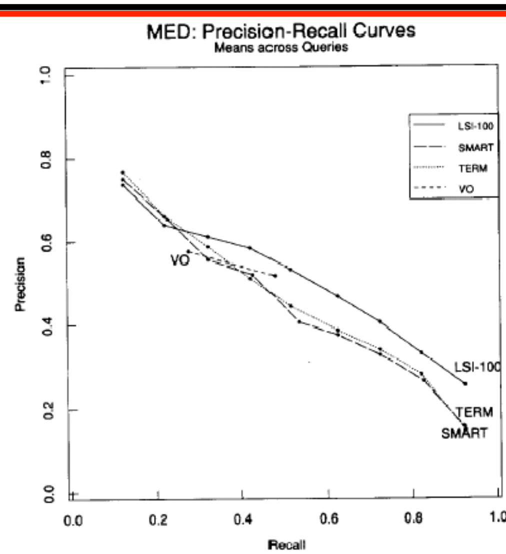
Documents were full text of title and abstract.

Stop list of 439 words (SMART); no stemming, etc.

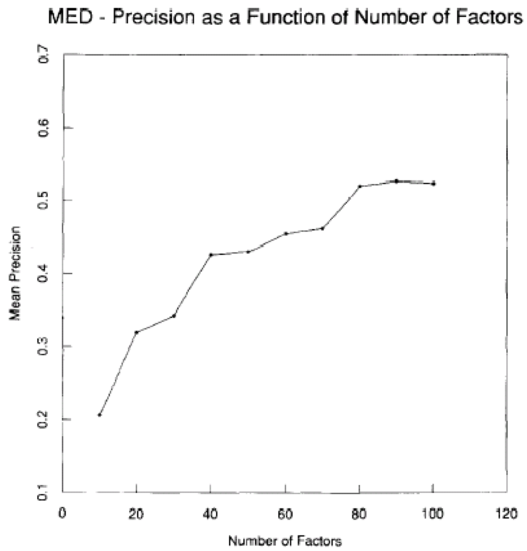
Comparison with:

(a) simple term matching, (b) SMART, (c) Voorhees method.

Experimental Results: 100 Factors



Experimental Results: Number of Factors (Concept)



Is LSI any good?

- Decomposes language into "basis vectors"
  - In a sense, is looking for core concepts
- In theory, this means that system will retrieve documents using synonyms of your query words
  - The "magic" that appeals to people
- Query "manna" on Bible verses (312 dimensions)

  #5 --Exodus 12_20 Ye shall eat nothing leavened; in all your habitations shall ye eat unleavened bread.

  #6 --Genesis 31_54 Then Jacob offered sacrifice upon the mount, and called his brethren to eat bread: and they did eat bread, and tarried all night in the mount.

Things like this are major claim of LSI techniques

Magic can be confusing

- Top 5 hits for query “apple” (312 dimensions)
  - *Song_of_Songs8_5* Who is this that cometh up from the wilderness, leaning upon her beloved? I raised thee up under the **apple** tree: there thy mother brought thee forth: there she brought thee forth that bare thee.
  - *Psalms 47_3* He shall subdue the people under us, and the nations under our feet. ????
  - Song_of_Songs2_3 As the **apple** tree among the trees of the wood, so is my beloved among the sons. I sat down under his shadow with great delight, and his fruit was sweet to my taste.
  - *Zecharaiah3_10* In that day, saith the LORD of hosts, shall ye call every man his neighbour under the vine and under the fig tree. Magic?
  - *Ecclesiastes 4_7* Then I returned, and I saw vanity under the sun. ????

http://lsi.research.telcordia.com/

Standard Vector Space vs LSI

- Standard vector space
  - Each dimension corresponds to a term in the vocabulary
  - Vector elements are real-valued, reflecting term importance
  - Any vector (document,query, ...) can be compared to any other
  - Cosine correlation is the similarity metric used most often
- Latent Semantic Indexing (LSI)
  - Each dimension corresponds to a “basic concept”
  - Documents and queries mapped into basic concepts
  - Same as standard vector space after that
  - Whether it’s good depends on what you want

Vector Space Model: Disadvantages

- Assumed independence relationship among terms – though this is a *very* common retrieval model assumption
- Lack of justification for some vector operations
  - e.g. choice of similarity function
  - e.g., choice of term weights
- Barely a retrieval model
  - Doesn't explicitly model relevance, a person's information need, language models, etc.
- Assumes a query and a document can be treated the same (symmetric)
- Lack of a cognitive (or other) justification

Vector Space Model: Advantages

- Simplicity
- Ability to incorporate term weights
  - *Any* type of term weights can be added
  - No model that has to justify the use of a weight
- Ability to handle "distributed" term representations
  - e.g., LSI
- Can measure similarities between almost anything:
  - documents and queries
  - documents and documents
  - queries and queries
  - sentences and sentences
  - etc.