LSA: From Information Retrieval to Team Performance

Dr. Melanie J. Martin
Mathematics and Computer Science Speaker Series
CSU Stanislaus
September 28, 2007

Outline of Talk

- Some Context
- A Tool for Information Retrieval
  - Latent Semantic Analysis
- More Than a Tool for Information Retrieval
  - Latent Semantic Analysis applied to Team Discourse
- Conclusion

First Some Context

- I am an Assistant Professor of Computer Science at CSU Stanislaus, I also teach Mathematics
- My dissertation title is "Reliability and Verification of Natural Language Text on the World Wide Web"
- This talk is about joint work with Peter W. Foltz at the Computing Research Laboratory at New Mexico State University

Areas I currently work in:

- Artificial Intelligence
  - Data Mining
  - Machine Learning
  - Natural Language Processing
- Information Retrieval
- Health Informatics
- Digital Media
- Computer Science Education

Information Retrieval

- Documents
- User has a need
- Find relevant content
- Indexing
  - How to represent
    - Documents
    - Requests (queries)
- Searching
  - How to match
    - Query to document
    - Based on indexing

http://www.screcordsmgmt.com/

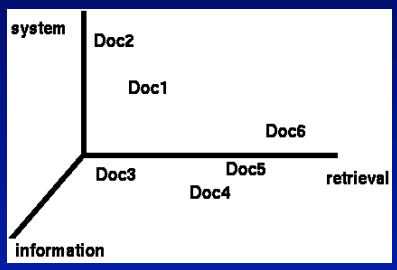
One way to do this:

- Vector Space Model (Salton et al. 1968)
- Represent a collection of documents
  - Term (rows) by document (columns) matrix, based on occurrence
  - Translate into vectors in a vector space
    - One dimension for each term (!)
    - Documents and queries are vectors of terms

Term by Context Matrix

Doc/ Term	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6
Information	0	0	1	1	1	0
Retrieval	0	0	1	1	1	1
System	1	1	1	0	0	0

Vector Space Model

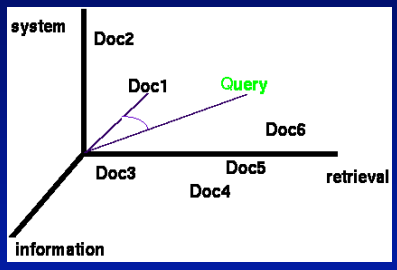


http://ei.cs.vt.edu/~cs5604/cs5604cnRR/RR-b1.html

One way to do this:

- Rank documents by how "close" they are to the query
  - "Close" is usually the computed using the cosine to measure distance between vectors (documents)
    - small angle = large cosine = similar
    - large angle = small cosine = dissimilar

Vector Space Model



http://ei.cs.vt.edu/~cs5604/cs5604cnRR/RR-b1.html

Some Minor Issues

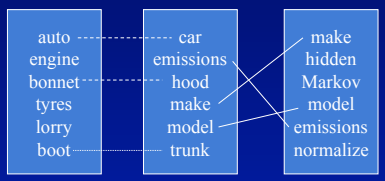
- What exactly are terms?
  - Stemming
  - Stop words
- Are all terms of equal importance?
  - Weighting schemes

Two Major Issues

- Words have multiple meanings (polysemy)
  - For example: model, python, chip, bank
- Multiple ways to refer to the same object or concept (synonymy)
  - For example: car-automobile

Two Major Issues

- Example: Vector Space Model
  - (from Lillian Lee)



Synonymy
Will have small cosine, but are related

Polysemy
Will have large cosine, but not truly related

Outline of Talk

- Some Context
- A Tool for Information Retrieval
  - Latent Semantic Analysis
- More Than a Tool for Information Retrieval
  - Latent Semantic Analysis applied to Team Discourse
- Conclusion

Latent Semantic Analysis (LSA)

- Latent Semantic Analysis was developed at Bellcore (now Telcordia) in the late 1980s (1988). It was patented in 1989.
- Goal
  - Address synonymy and polysemy issues
  - Model underlying/hidden/latent semantic relationship between terms and documents
  - Things with same meaning should be close together in the space

Latent Semantic Analysis

- Strategy
  - Reduce dimension of space
    - Semantically related terms end up in same dimension
  - Exploit term co-occurrence
    - Two or more terms occur in same documents more often than chance
    - If words co-occur in similar contexts => evidence of semantic relatedness
      - D1: user interface HCI interaction
      - D2: HCI interaction
      - Query: user interface

LSA and Co-occurrence

- LSA operates on the deep level (latent) meaning of words rather than the surface characteristics (exact matches). Example:
  - The doctor operates on the patient.
  - The physician is in surgery.
- No term overlap

	Term	LSA
– Doctor—Doctor	1.0	1.0
– Doctor—Physician	0.0	0.8
– Doctor—Surgeon	0.0	0.7

Representation of "Life"

LIFE:

```
 0.03992  -0.023    0.01394  -0.006452 -0.04221  -0.05371
-0.001017 -0.04663  -0.0228   -0.02284  -0.0542    0.02118
 0.02703   0.04236   0.013     0.0151   -0.01519   0.02841
-0.02139  -0.0252    0.002628 -0.007437 -0.003438  0.01788
-0.05929   0.02553  -0.01334   0.02155   0.009091  0.03491
-0.005196  0.009027 -0.001789 -0.04187   0.006131  0.005329
 0.0114   -0.01655   0.01126   0.05759  -0.04004  -0.01597
 0.0301    0.001113 -0.02021   0.02676   0.003837  0.0003557
 0.0495    0.04604  -0.004699  0.008017 -0.06222  -0.05293
-0.009894  0.00355  -0.005064 -0.01819  -0.004684  0.01215
-0.04272   0.008417  0.04143  -0.001864 -0.02142   0.01003
-0.02885  -0.003961 -0.0143    0.02333  -0.000221 -0.02247
 0.02521  -0.02099  -0.01562   0.02417  -0.005734  0.001533
-0.008991  0.01218  -0.01653  -0.008191 -0.006373 -0.03939
-0.002844 -0.002278 -0.01121  -0.05195  -0.01264  -0.001516
 0.03375   0.01118   0.02304  -0.03583   0.03462   0.04268
-0.02618   0.009468  0.01484  -0.007926 -0.03572  -0.02196
-0.03567  -0.04822   0.02627  -0.001668  0.004044  0.007416
 0.005181 -0.03249  -0.0165    0.01675   0.007954 -0.01064
-0.03624   0.00626  -0.023     0.04962   0.0392    0.008223
-0.002816  0.03494   0.003373  0.04208   0.02945  -0.07585
-0.003087  0.03298  -0.02058   0.000470  0.03793  -0.02148
-0.04032   0.01518  -0.01361   0.02362  -0.0008575 0.03437
 0.02592   0.01731  -0.06542  -0.02625  -0.009007  0.01611
```

Meaning of "Life"

1	0.62	lives	47	0.45	other
2	0.58	lifetime	48	0.45	soul
3	0.56	everyday	52	0.44	sustain
4	0.55	death	53	0.43	pattison
5	0.54	living	57	0.43	almost
6	0.52	loneliness	58	0.43	comfort
8	0.50	bereavement	63	0.43	own
11	0.50	pleasures	64	0.43	who
13	0.49	alive	65	0.42	born
14	0.49	despair	67	0.42	on
15	0.49	dying	68	0.42	no
16	0.49	existence	69	0.42	challenge
20	0.48	happiness	70	0.42	eternity
22	0.48	die	74	0.42	years
23	0.48	dies	75	0.42	beyond
24	0.48	comforts	76	0.42	becoming
27	0.47	totally	77	0.42	throughout
28	0.47	span	79	0.41	loving
30	0.47	worlds	80	0.41	childhood
32	0.47	desire	81	0.41	surroundings
35	0.46	time	82	0.41	because
37	0.46	inevitability	83	0.41	everywhere
42	0.46	romanticizing	85	0.41	reincarnation
43	0.45	dehumanized	86	0.41	companionship

Latent Semantic Analysis

- Implementation: Four Basic Steps
  - Term by document (context) matrix
  - Convert matrix entries to weights
  - Singular Value Decomposition (SVD) performed on matrix
  - Reduce Rank of matrix
    - all but the k highest singular values are set to 0
    - produces k-dimensional approximation of the original matrix (in least-squares sense)
    - this is the "semantic space"

Latent Semantic Analysis

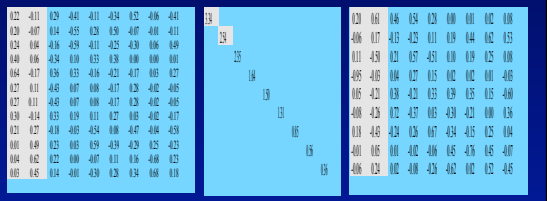
- Singular Value Decomposition

$$X = T_0 S_0 D_0^T$$

- Dimension Reduction

$$\hat{X} = TSD^T$$

A Small Example - 3



T = terms x dimensions S = singular values D = documents x dimensions

Started with 12 terms and 9 documents

Latent Semantic Analysis

- A theoretical model of cognitive phenomena
- A practical tool for measuring cognitive artifacts based on semantic information
- Provides measures of the semantic relatedness, quality, and quantity of information contained in discourse
- Automatic and fast

Summary

- Some Issues
  - SVD Algorithm complexity O(n^2*k^3)
    - n = number of terms
    - k = number of dimensions in semantic space (typically small ~50 to 350)
    - for stable document collection, only have to run once
    - not so good for large, dynamic document collection

Summary

- Some issues
  - Finding optimal dimension for semantic space
    - precision-recall improve as dimension is increased until hits optimal, then slowly decreases until it hits standard vector model
    - don't want to run SVD too may times to find optimal dimension
    - research being done on this

Summary

- Some issues
  - SVD assumes normally distributed data
    - term occurrence is not normally distributed
    - matrix entries are weights, not counts, which may be normally distributed even when counts are not