

Latent Semantic Indexing: What can we learn from a close study of the data?

April Kontostathis

Ursinus College

October 8, 2007



URSINUS

What is Information Retrieval?

- **finding**, accessing, and downloading **digital information** through networks.
- On the WWW, the most important method of IR is the **indexing of free-form text**. IR exhibits similarities to (but is not the same as) other areas of information processing, such as expert systems and data base management systems (DBMS).
- **All** of the **combined** series of screens of **information** available to a Telephone Secretary or a Dispatcher regarding an account, including call-processing procedure, phone numbers and other contact information for key individuals, directions or other instructions which may be given to a caller, definition of an emergency, emergency procedures, etc.
- The activity of retrieving information by **extracting documents** or its parts from larger quantities of documents with the help of a computer, auxiliary structures and mathematical methods.
- IR is the process of **determining the relevant documents** from a collection of documents, based on a query presented by the user.



Why do we care?

- **Information overload** - To characterize documents by topic, with little or no human intervention, to help filter information
- **Query Processing** - To answer *ad hoc* questions
- **Discovery** - To identify new interesting trends in documents and corpora



What is Latent Semantic Indexing?

Method for automatic indexing and retrieval of documents within a text collection.

Uses the matrix of observed occurrences to estimate the underlying semantic model. i.e. reveal the **latent semantics** present in the collection.



Matrix of observed occurrence (term by document matrix)

- 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

Deerwester Term by Document Matrix

	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



Latent Semantic Indexing (LSI)

- Singular Value Decomposition (SVD) is used to decompose term doc matrix
- Resulting Matrices are truncated to k dimensions
- “Captures the most important underlying structure in the association of terms and documents, yet at the same time removes the noise or variability in word usage that plaques word-based retrieval methods.”
- Use the interrelationships among terms to improve retrieval.

Berry, M.W., S.T. Dumais, and G.W. O'Brien. 1995. Using linear algebra for intelligent information retrieval. *SIAM Review*, Volume 37, No. 4. pp. 573-595.



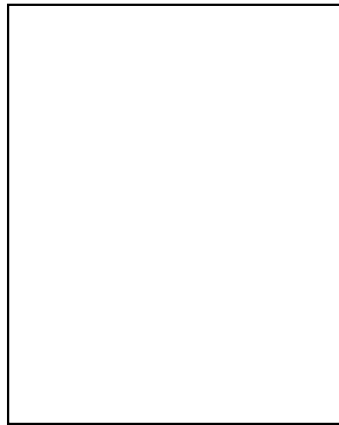
Useful Characteristics of Matrix Decomposition Techniques

- Incorporate Higher-order Correlation Information
- Can Exploit Auxiliary Information Associated with both Edges and Nodes of Networks
- Are Robust in the Presence of Missing and Incorrect Data
- Scale much Better than many Tools Classically used in Social Network Analysis and Link Analysis

D. Skillicorn, Social Network Analysis via Matrix Decompositions, *in*: Emergent Information Technologies and Enabling Policies for Counter-Terrorism, Wiley, 2006



Singular Value Decomposition



$A (m \times n)$

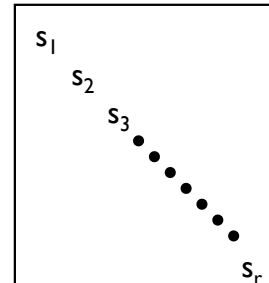
Original
Term by Doc

=



$T (m \times r)$

Term by
Dimension



$S (r \times r)$

Singular
Values

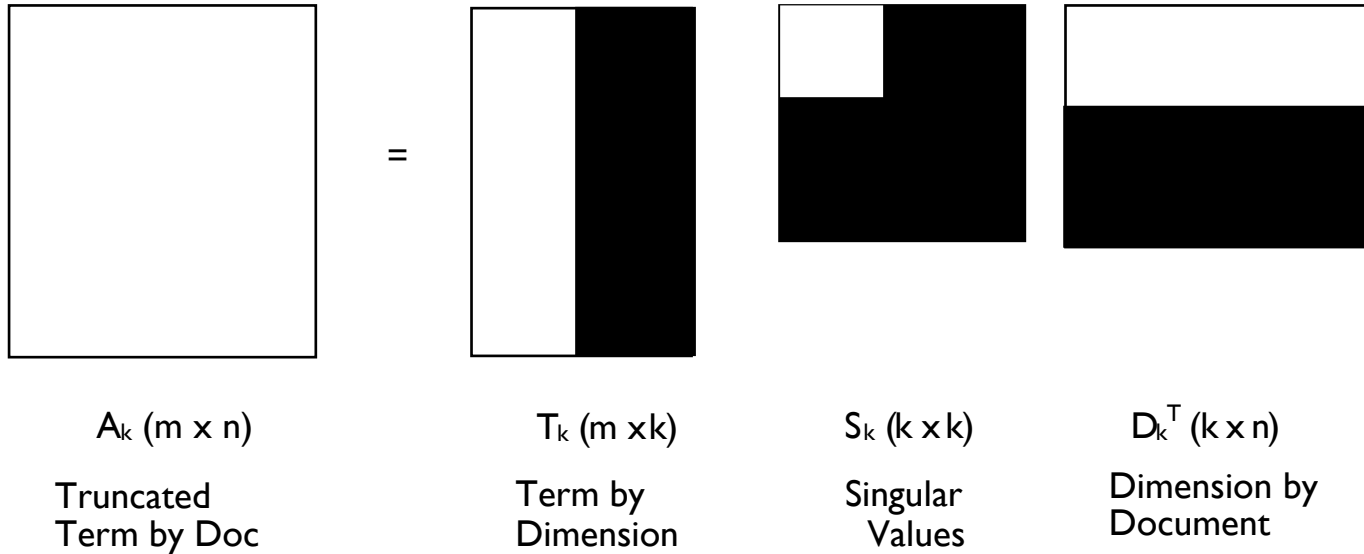


$D^T (r \times n)$

Dimension by Document



Truncated SVD



Deerwester et al., Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science*, 41(6), pp. 391-407, October, 1990.



Example Factorization

Deerwester Term by Document Matrix									
	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

Deerwester Term by Document Matrix, Truncated to Two Dimensions									
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.37	0.46	0.18	(0.05)	(0.12)	(0.16)	(0.09)
interface	0.14	0.38	0.33	0.40	0.17	(0.03)	(0.07)	(0.10)	(0.04)
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.05	0.08	0.12
user	0.26	0.84	0.59	0.69	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.04	1.25	0.55	(0.08)	(0.17)	(0.22)	(0.06)
response	0.16	0.60	0.38	0.42	0.28	0.05	0.13	0.19	0.22
time	0.16	0.60	0.38	0.42	0.28	0.05	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.62	0.24	(0.07)	(0.15)	(0.20)	(0.11)
Survey	0.10	0.54	0.23	0.22	0.27	0.13	0.31	0.44	0.42
trees	(0.07)	0.23	(0.15)	(0.27)	0.15	0.24	0.55	0.77	0.66
graph	(0.07)	0.35	(0.14)	(0.29)	0.21	0.30	0.69	0.98	0.85
minors	(0.05)	0.26	(0.10)	(0.21)	0.15	0.22	0.50	0.71	0.61



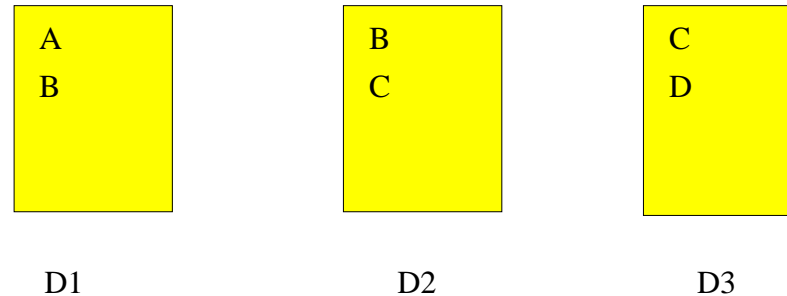
Question: Why does LSI Work?

- Answer: Because it captures co-occurrence relationships between terms and assigns either a positive or negative similarity value

Kontostathis, April and William M. Pottenger. (2006) A framework for understanding LSI performance. Information Processing and Management. Volume 42, number 1, pages 56-73.



What do we mean by Term Co-occurrence?



First Order Co-occurrence	{A,B}, {B,C}, {C,D}
Second Order Co-occurrences	{A,C}, {B,D}
Third Order Co-occurrences	{A,D}

LSI and Higher Order Co-occurrence

1st Order: computer → interface = .52

2nd Order: user → interface → human = .94

3rd Order: trees → graph → survey → computer = .15

3rd Order: graph → survey → computer → human = -.34

	human	interface	computer	user	system	response	time	EPS	Survey	trees	graph	minors
human	x	1	1	0	2	0	0	1	0	0	0	0
interface	1	x	1	1	1	0	0	1	0	0	0	0
computer	1	1	x	1	1	1	1	0	1	0	0	0
user	0	1	1	x	2	2	2	1	1	0	0	0
system	2	1	1	2	x	1	1	3	1	0	0	0
response	0	0	1	2	1	x	2	0	1	0	0	0
time	0	0	1	2	1	2	x	0	1	0	0	0
EPS	1	1	0	1	3	0	0	x	0	0	0	0
Survey	0	0	1	1	1	1	1	0	x	0	1	1
trees	0	0	0	0	0	0	0	0	0	x	2	1
graph	0	0	0	0	0	0	0	0	1	2	x	2
minors	0	0	0	0	0	0	0	0	1	1	2	x

	human	interface	computer	user	system	response	time	EPS	Survey	trees	graph	minors
human	x	0.54	0.56	0.94	1.69	0.58	0.58	0.84	0.32	-0.32	-0.34	-0.25
interface	0.54	x	0.52	0.87	1.50	0.55	0.55	0.73	0.35	-0.20	-0.19	-0.14
computer	0.56	0.52	x	1.09	1.67	0.75	0.75	0.77	0.63	0.15	0.27	0.20
user	0.94	0.87	1.09	x	2.79	1.25	1.25	1.28	1.04	0.23	0.42	0.31
system	1.69	1.50	1.67	2.79	x	1.81	1.81	2.30	1.20	-0.47	-0.39	-0.28
response	0.58	0.55	0.75	1.25	1.81	x	0.89	0.80	0.82	0.38	0.56	0.41
time	0.58	0.55	0.75	1.25	1.81	0.89	x	0.80	0.82	0.38	0.56	0.41
EPS	0.84	0.73	0.77	1.28	2.30	0.80	0.80	x	0.46	-0.41	-0.43	-0.31
Survey	0.32	0.35	0.63	1.04	1.20	0.82	0.82	0.46	x	0.88	1.17	0.85
trees	-0.32	-0.20	0.15	0.23	-0.47	0.38	0.38	-0.41	0.88	x	1.96	1.43
graph	-0.34	-0.19	0.27	0.42	-0.39	0.56	0.56	-0.43	1.17	1.96	x	1.81
minors	-0.25	-0.14	0.20	0.31	-0.28	0.41	0.41	-0.31	0.85	1.43	1.81	x



Disjoint Collection

Deerwester Term by Document Matrix

	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

Modified Deerwester, term by term, truncated to 2 dimensions

	human	interface	computer	user	system	response	time	EPS	Survey	trees	graph	minors
human	0.56	0.50	0.60	1.01	1.62	0.66	0.66	0.76	0.45	-	-	-
interface	0.50	0.45	0.53	0.90	1.45	0.59	0.59	0.68	0.40	-	-	-
computer	0.60	0.53	0.64	1.08	1.74	0.71	0.71	0.81	0.48	-	-	-
user	1.01	0.90	1.08	1.82	2.92	1.19	1.19	1.37	0.81	-	-	-
system	1.62	1.45	1.74	2.92	4.70	1.91	1.91	2.20	1.30	-	-	-
response	0.66	0.59	0.71	1.19	1.91	0.78	0.78	0.90	0.53	-	-	-
time	0.66	0.59	0.71	1.19	1.91	0.78	0.78	0.90	0.53	-	-	-
EPS	0.76	0.68	0.81	1.37	2.20	0.90	0.90	1.03	0.61	-	-	-
Survey	0.45	0.40	0.48	0.81	1.30	0.53	0.53	0.61	0.36	-	-	-
trees	-	-	-	-	-	-	-	-	-	2.05	2.37	1.65
graph	-	-	-	-	-	-	-	-	-	2.37	2.74	1.91
minors	-	-	-	-	-	-	-	-	-	1.65	1.91	1.33

Remove this Value



Theorem

If the ij^{th} element of the truncated term by term matrix, Y , is nonzero, then there is a connectivity path between term i and term j .

Alternately, we can say that if there is no path between terms i and j , then $y_{ij} = 0$ for all k .

Kontostathis, April and William M. Pottenger. (2006) A framework for understanding LSI performance. Information Processing and Management. Volume 42, number 1, pages 56-73.



Observation

- Positive, negative and 'near zero' values occur in the term-term matrix for all orders of co-occurrence



Question

- Is there a relationship between the distribution of values in the SVD matrices and LSI Performance?
 - We tried removing values to determine the impact on performance
 - High Positive
 - High Negative
 - Near Zero



Considerations

- Negative values are important
 - We tried to remove all negatives and retrieval performance decreased dramatically
 - We decided to keep an equal number of positive and negative values
- We want to balance the information provided by the term and document matrices
 - If the value in element (i, j) of the term matrix represents the importance of term i in dimension j , then the value in element (i, j) of the document matrix represents the importance of document i in dimension j



Sparsification Algorithm

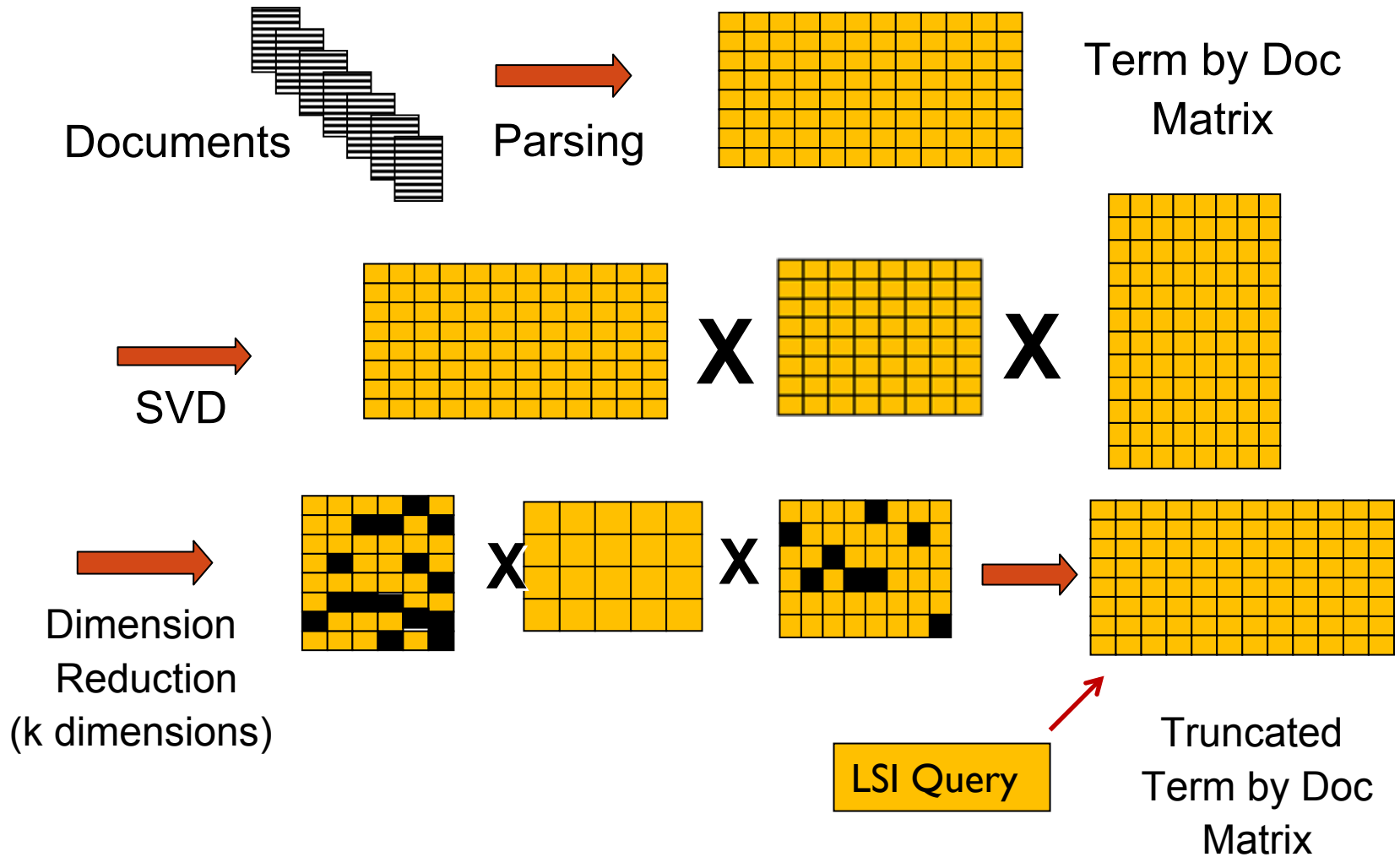
Identify threshold values that result in removal of a fixed percentage of the values in the term by dimension matrix

Use this threshold to truncate both the term matrix and the document matrix

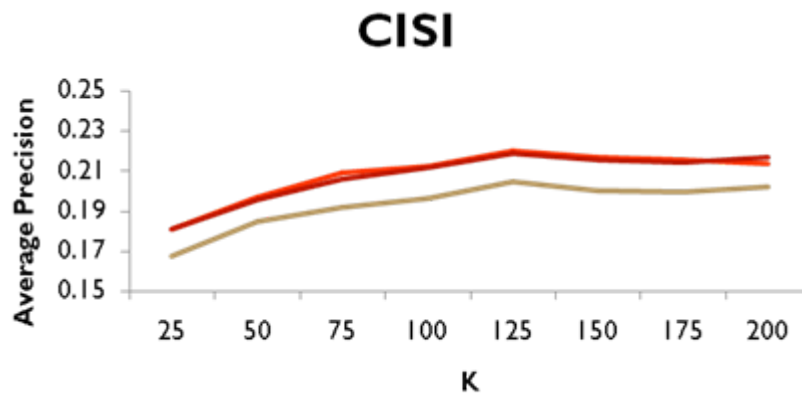
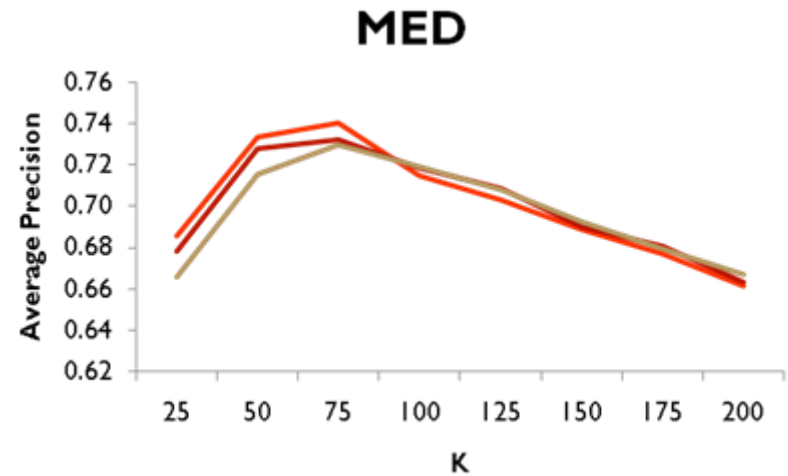
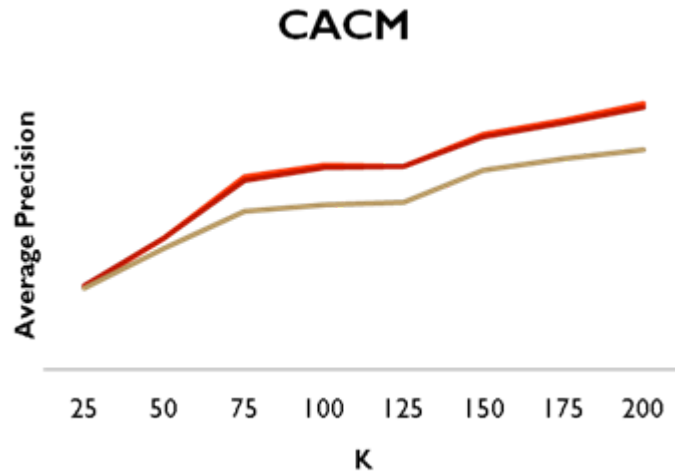
Kontostathis, April, William M. Pottenger, and Brian D. Davison. (2004) Assessing the impact of sparsification on LSI Performance. Proceedings of the 2004 Grace Hopper Celebration of Women in Computing Conference. Oct 6-9, 2004. Chicago, IL



Information Retrieval Process



Retrieval Performance – Small corpora

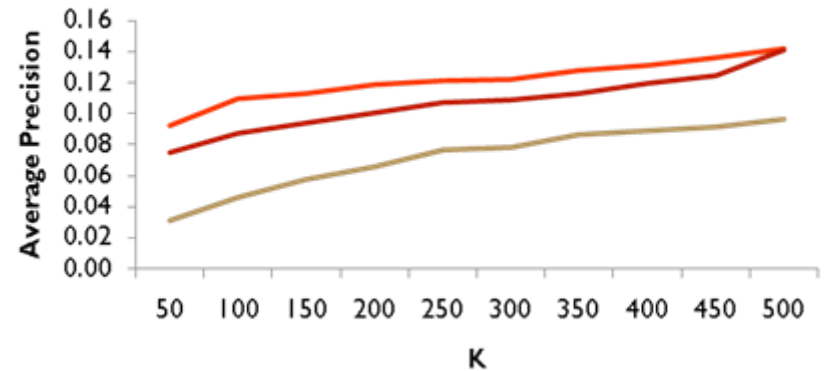


- LSI Baseline
- 70% Sparse
- 90% Sparse

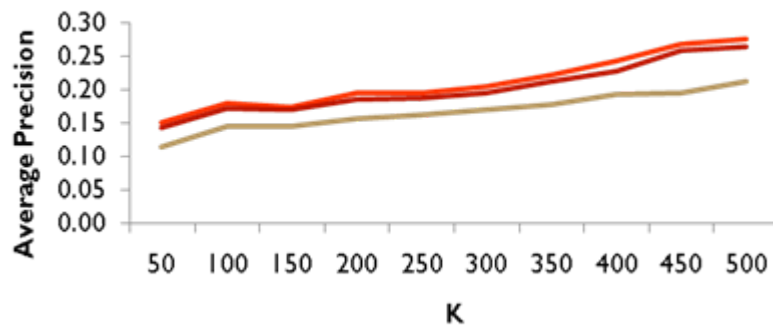


Retrieval Performance – Medium Corpora

NPL



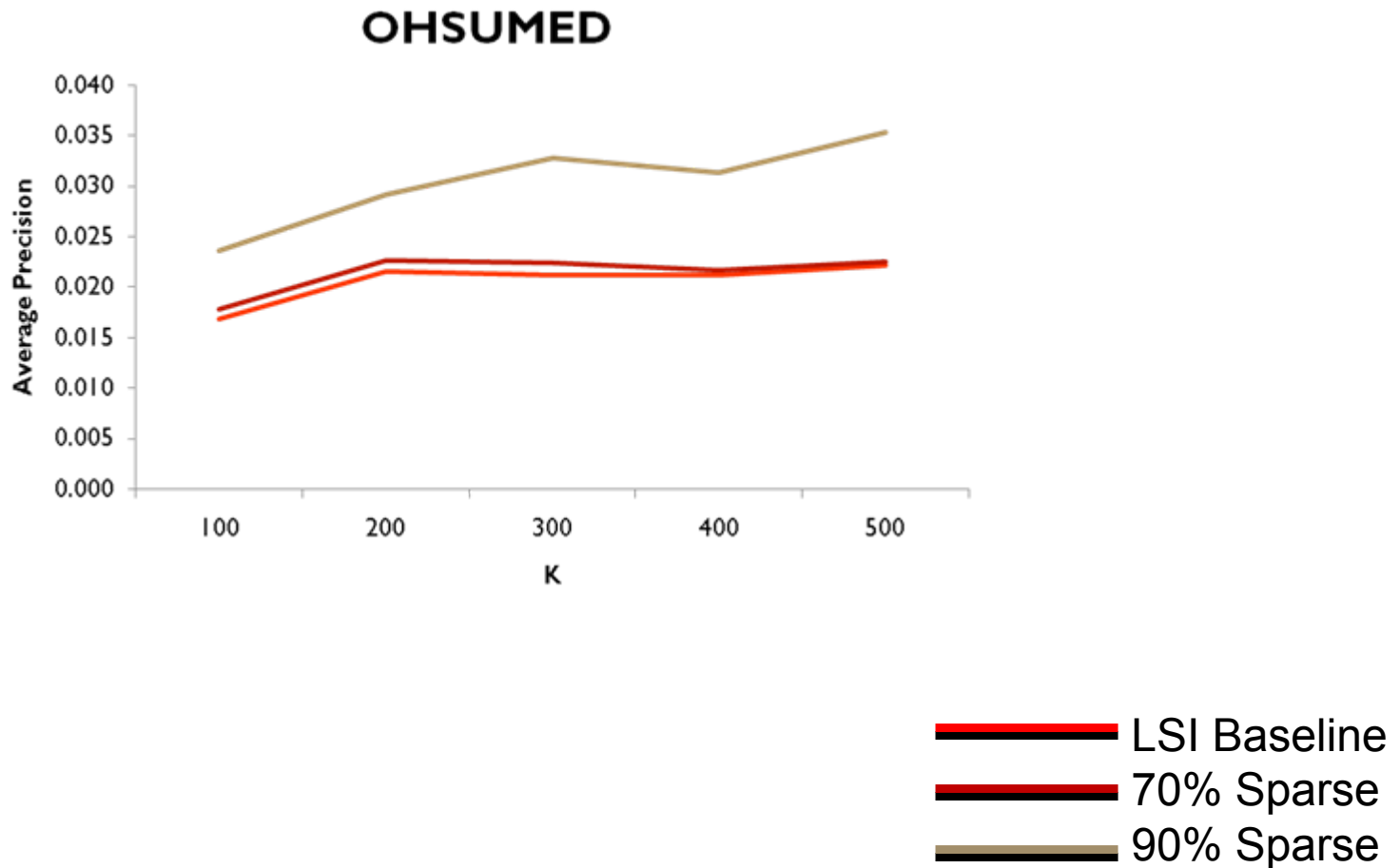
LISA



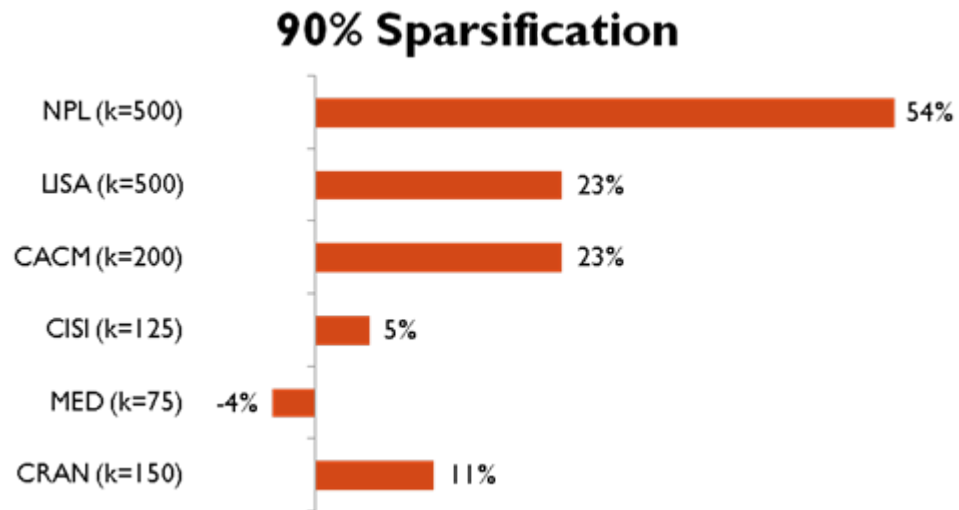
- LSI Baseline
- 70% Sparse
- 90% Sparse



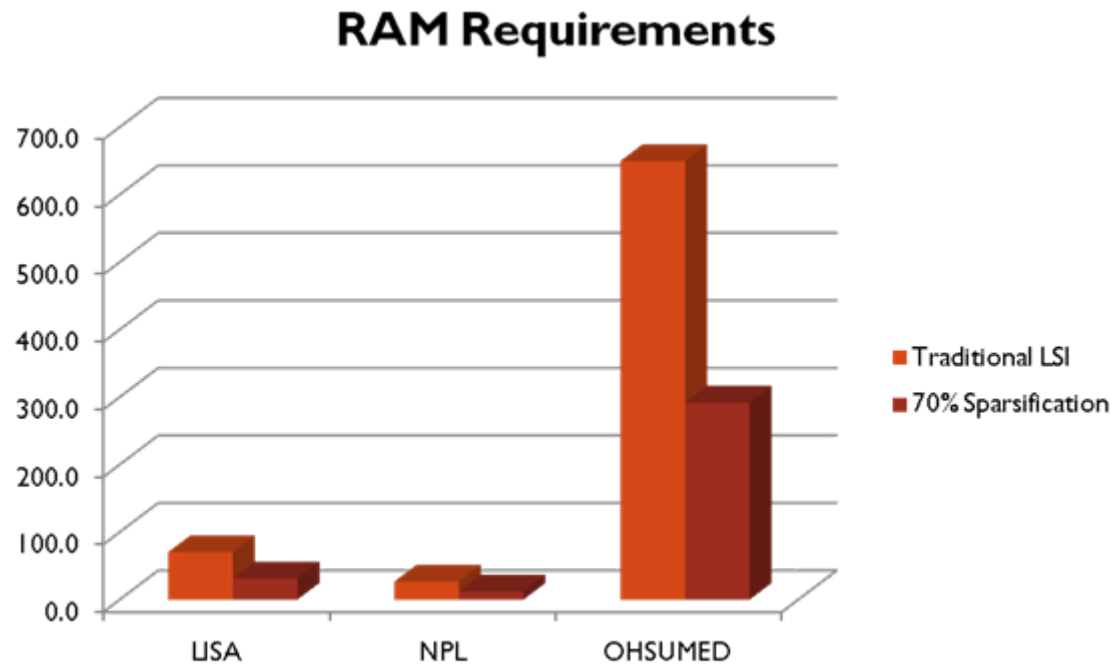
Retrieval Performance – Large Corpus



Query Run Time Improvement with Sparsification



Reduced RAM Requirements

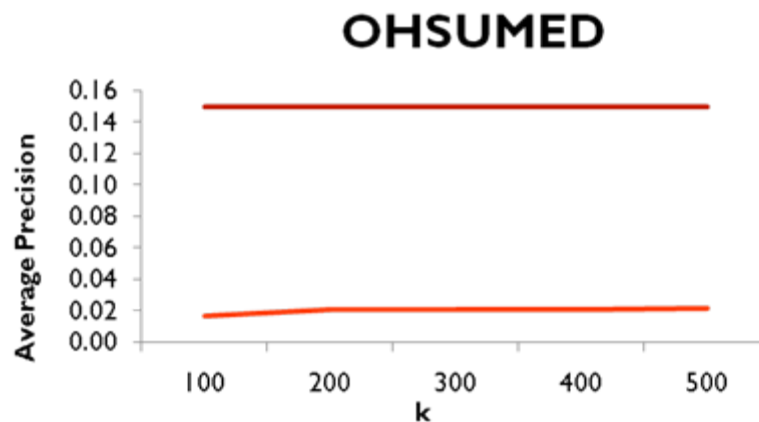
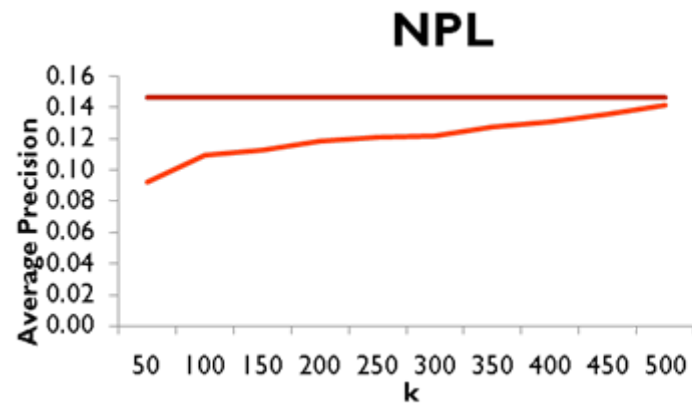
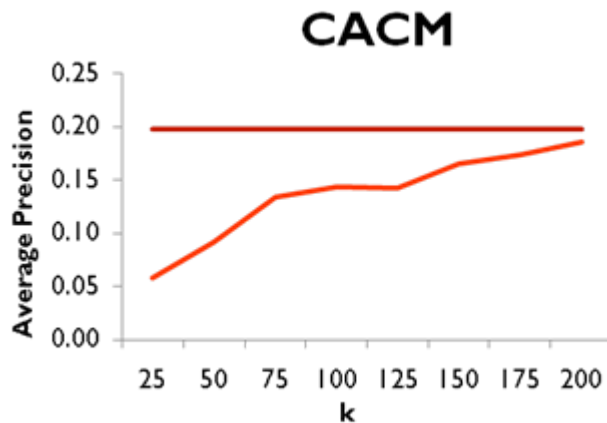


Problems with LSI

- Computing SVD computationally expensive for even moderately sized collections of documents
 - SVD approximation algorithms have been developed
- Term and Document vectors are much denser than the original term by document resulting in slow query run times, large RAM requirements
 - Sparsification and compression techniques have been developed
- ➔ LSI sometimes hurts retrieval!
- ➔ Optimal K is hard to identify, especially for large collections



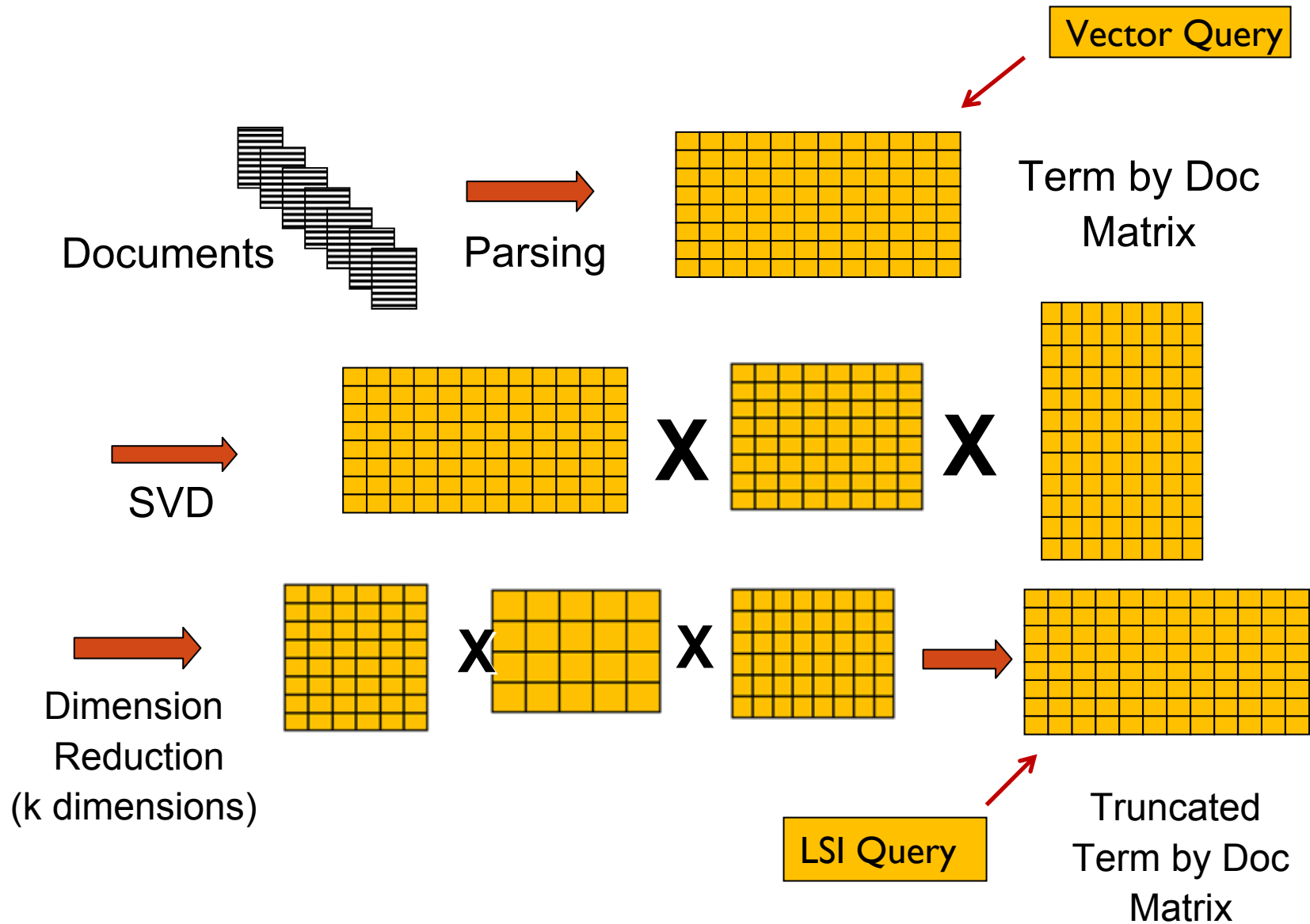
LSI compared to Traditional Vector Space Retrieval



— LSI
— Vector Space

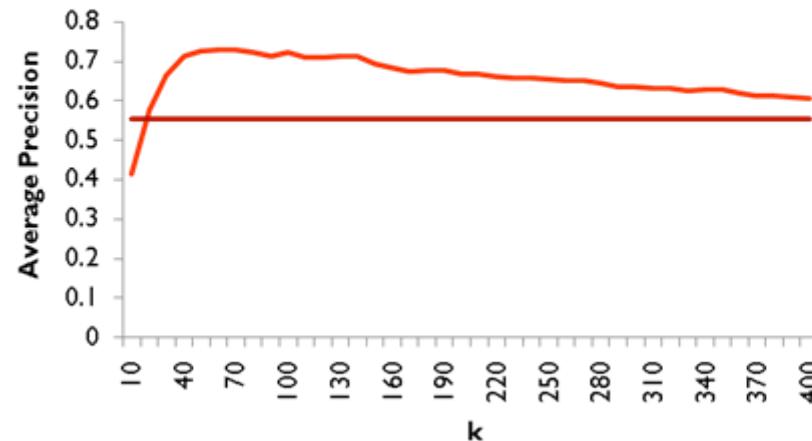


Information Retrieval Process



Notice LSI performance as k increases

**Average Precision for LSI and Vector
MED Corpus, Rank = 1033**



In fact, LSI is equivalent to vector space retrieval as k approaches the rank of the original term by document matrix.



Question: How are the ‘latent semantics’ represented?

- Answer: We have evidence that shows that the **term relationship** information is captured within the first few dimensions.

Kontostathis, April. (2007) Essential Dimensions of Latent Semantic Indexing (EDLSI). Proceedings of the 40th Annual Hawaii International Conference on System Sciences (CD-ROM). January 2007. Computer Society Press.



Essential Dimensions of LSI (EDLSI)

- Use the term relationship information, captured in the first few SVD vectors, in combination with traditional vector space retrieval.
 - Fewer SVD dimensions need to be computed
 - Fewer dense vectors in the query process – reduced RAM requirements, faster query runtime performance
 - Process is not as sensitive to changes in \mathbf{k}



EDLSI




- Document weight vector, w , defined to be

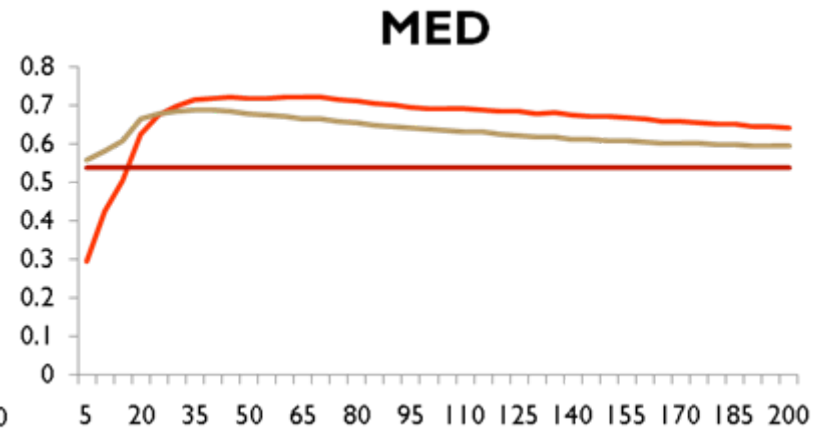
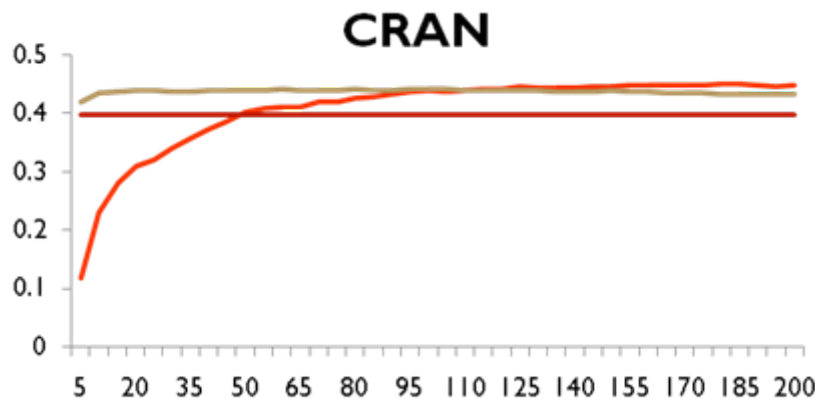
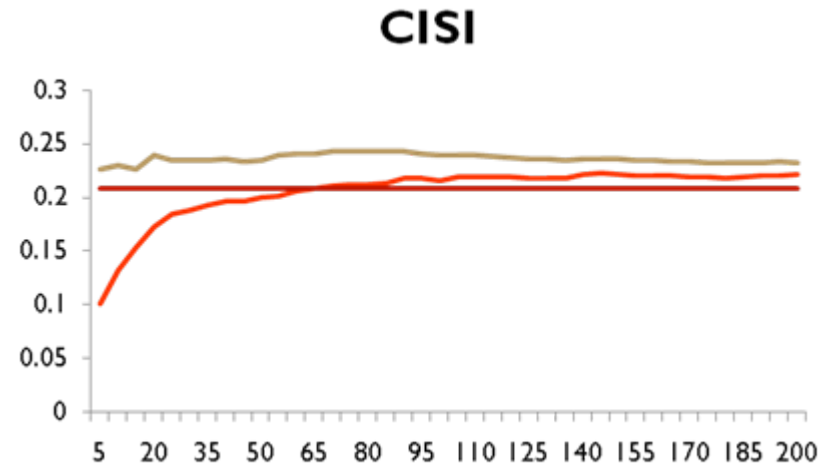
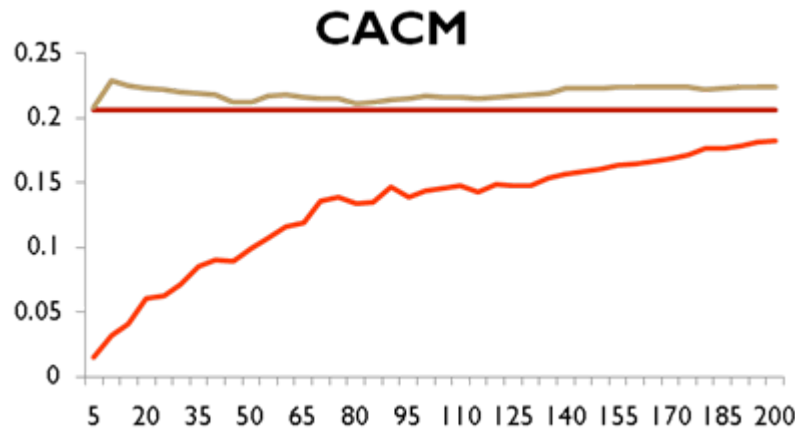
$$w = (x)(qA_k) + (1 - x)(qA)$$

- x is a weighting factor ($0 \leq x \leq 1$) that determines how much weight should be given to LSI vs. Vector retrieval
- k is small



EDLSI (LSI contributing 20%) Small Corpora

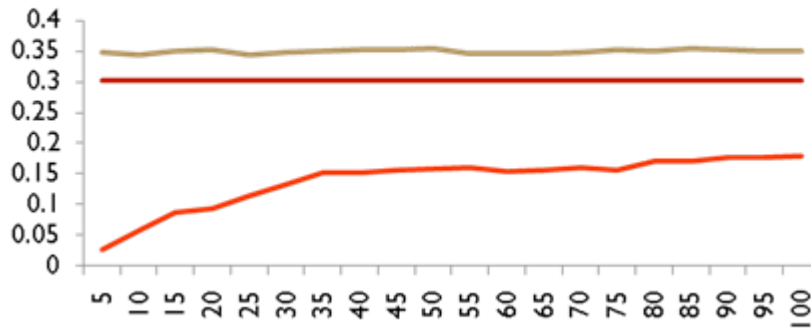
 LSI
 Vector
 EDLSI



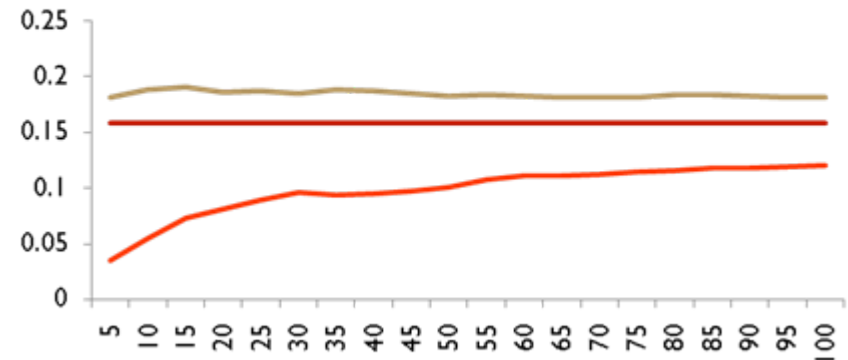
EDLSI (LSI contributing 20%) Midsize Corpora

LSI
Vector
EDLSI

LISA



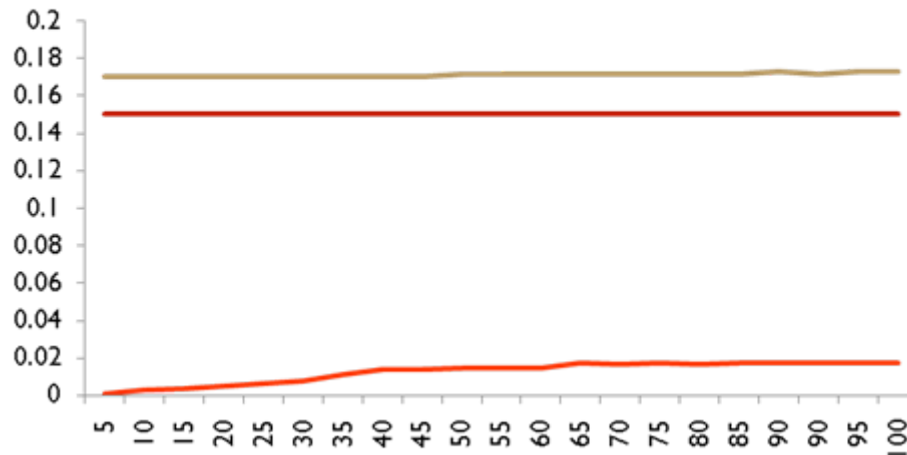
NPL



EDLSI (LSI contributing 20%) Large Corpus

LSI
Vector
EDLSI

OHSUMED



Applications of LSI – Emerging Trend Detection

An emerging trend is a topic area that has recently appeared in the literature and is growing in interest and utility over time.

Year	Number of documents
1994	3
1995	1
1996	8
1997	10
1998	170
1999	371

Emergence of XML in the mid-1990s



Applications - ETD

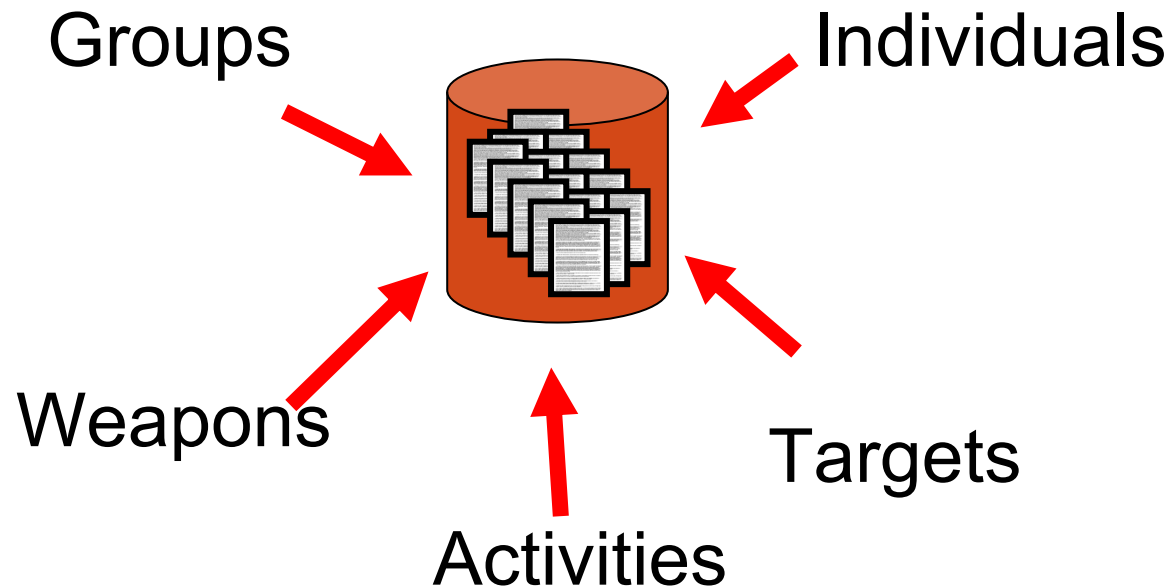
- LSI-based term clusters can be used to predict emerging trends more effectively than other clustering techniques, resulting in higher precision and recall rates.

Collection	Recall	Precision	F-Measure
Inspecc® 96	.89	.65	.75
Inspecc® 97	.97	.55	.70
Inspecc® 98	.96	.64	.77
Inspecc® 99	.93	.62	.74
Average	.94	.62	.74

Kontostathis, April, Indro De, Lars E. Holzman and William M. Pottenger. (2004) [Use of Term Clusters for Emerging Trend Detection](#). Technical Report.



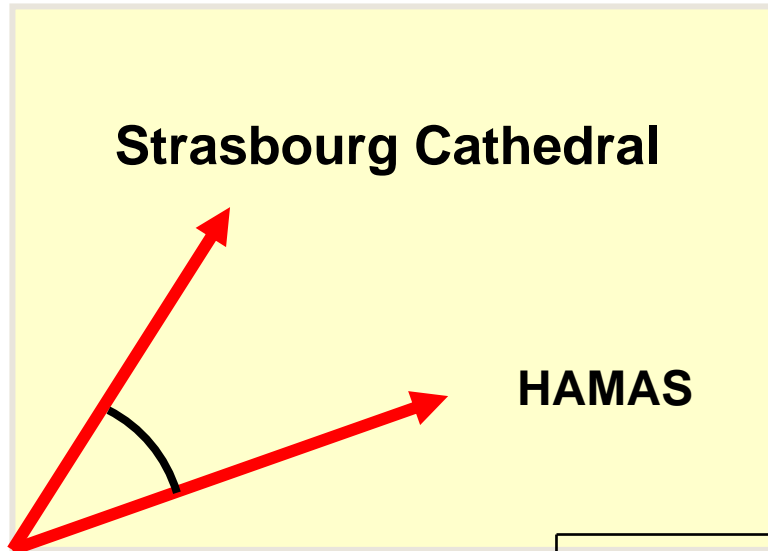
Applications of LSI – Discovering Entity Relationships



R.B. Bradford. (2006) Relationship Discovery in Large Text Collections Using Latent Semantic Indexing. Proceedings of the 2006 SIAM Conference on Data Mining Workshop on Link Analysis, Counterterrorism and Security



Application – Entity relationships



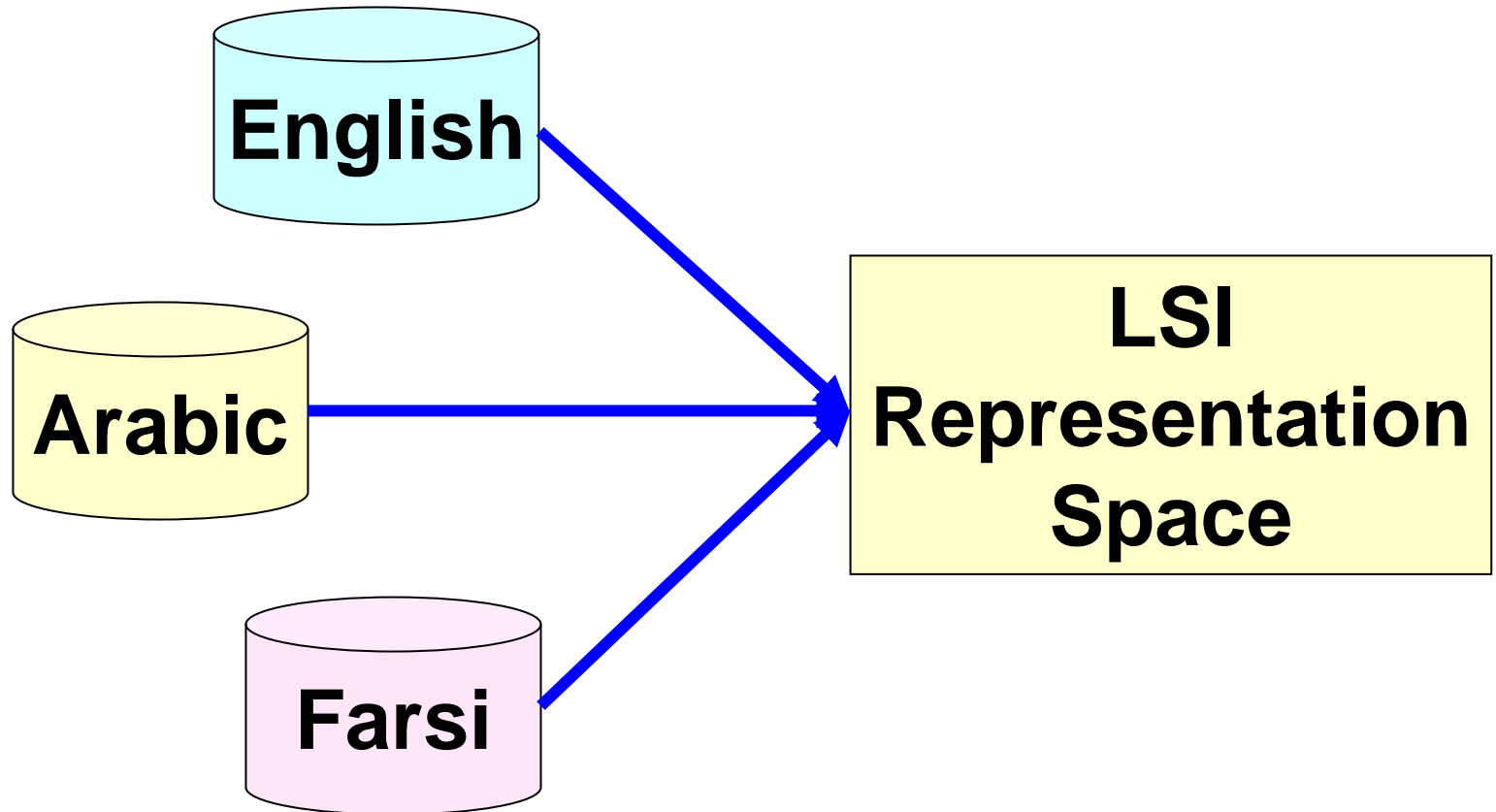
Cosine Distance
between vectors in
LSI space

No document mentions both GSPC (or any synonym) and Strasbourg Cathedral, but an attack was planned

	Athens Airport	Strasbourg Cathedral	Trafalgar Square
Abu Sayyaf Group	-.0254	-.1235	-.0862
GSPC	-.1556	.5087	.1677
HAMAS	.0071	-.0019	.0152
PFLP	.0580	.0101	.0931
Salafia Jihadia	.0101	.2928	-.0264



Applications – Multilingual retrieval/translation



Other Applications of LSI

- **Author Identification**

Ian M. Soboroff, Charles K. Nicholas, James M. Kukla, and David S. Ebert. Visualizing Document authorship using n-grams and latent semantic indexing. In Proceedings of the Workshop on New Paradigms in Information Visualization and Manipulation (NPIV '97), Las Vegas, NV, USA, November 1997.

- **Image Retrieval**

Latent Semantic Indexing for Image Retrieval Systems, SIAM LA
P Praks, V Snasel, J Dvorsky - International Linear Algebra Society 2003

- **Discourse Analysis**

M. J. Martin and P. W. Foltz. Automated team discourse annotation and performance prediction using LSA. HLT-NAACL 2004: pages 97–100



Conclusions

- LSI captures higher order term relationship information
- Some entries in the SVD matrices can be removed to reduce RAM requirements
- A fusion between LSI and Vector Space retrieval provides a robust method for retrieval across a variety of disparate collections
- LSI remains a relevant option for many applications



Contact Information

April Kontostathis

Ursinus College

akontostathis@ursinus.edu

<http://webpages.ursinus.edu/akontostathis>

610-409-3000 x2650



URSINUS

April Kontostathis
Department of Mathematics and Computer Science