Carnegie Mellon

15-826: Multimedia Databases and Data Mining

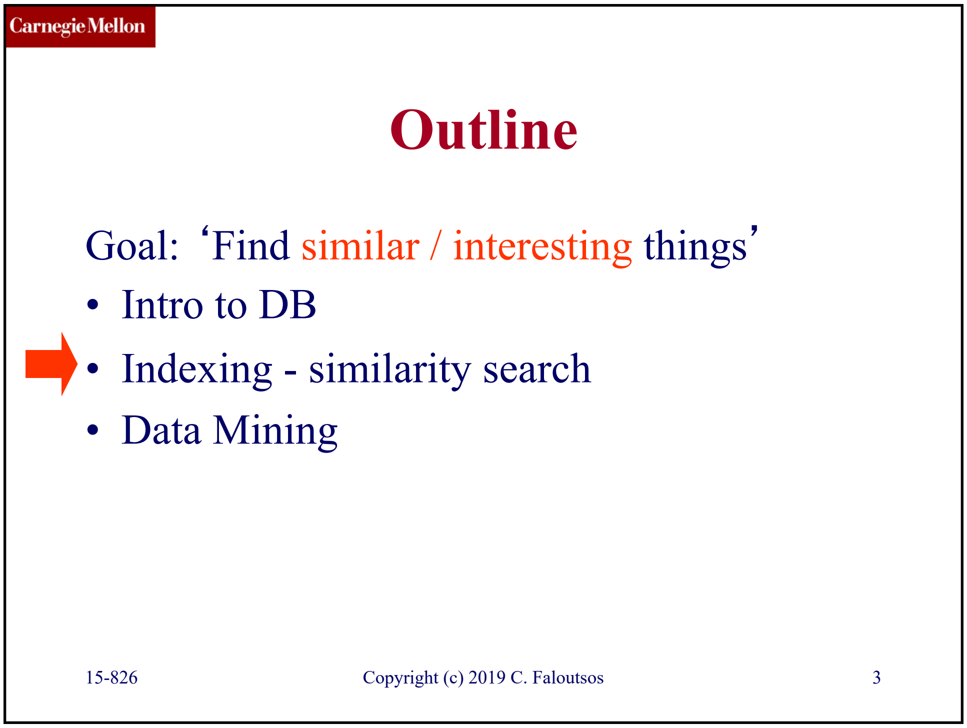
Lecture #15: Text - part IV (LSI)

C. Faloutsos

Carnegie Mellon

Must-read Material

- Foltz, P. W. and S. T. Dumais (Dec. 1992). "Personalized Information Delivery: An Analysis of Information Filtering Methods." Comm. of ACM (CACM) 35(12): 51-60.



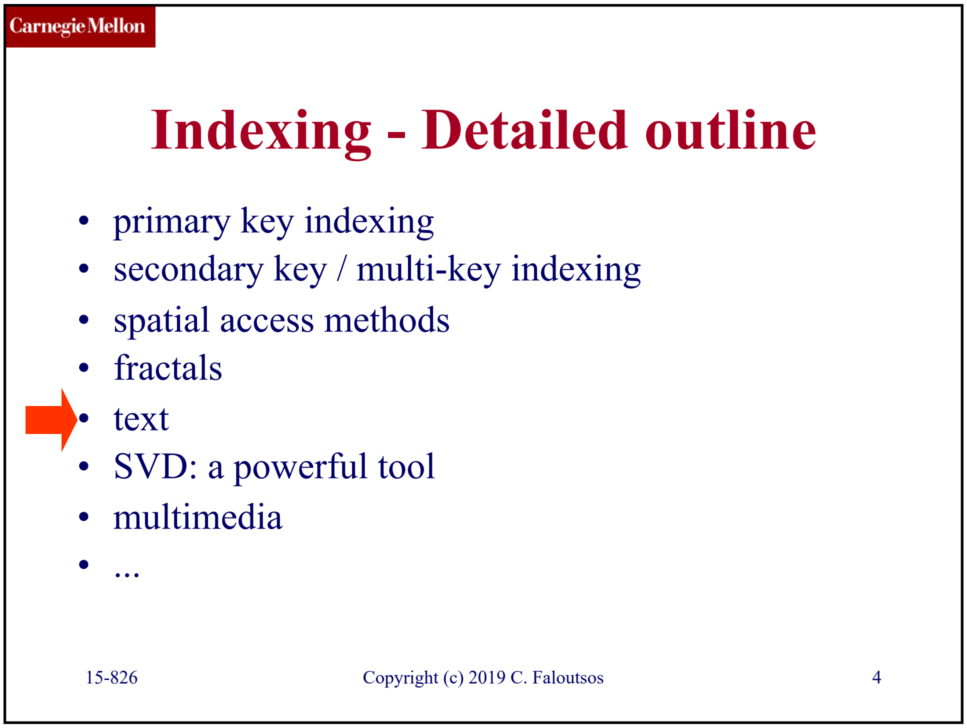
Carnegie Mellon

Outline

Goal: ‘Find similar / interesting things’

- Intro to DB
- ➡ Indexing - similarity search
- Data Mining

15-826 Copyright (c) 2019 C. Faloutsos 3

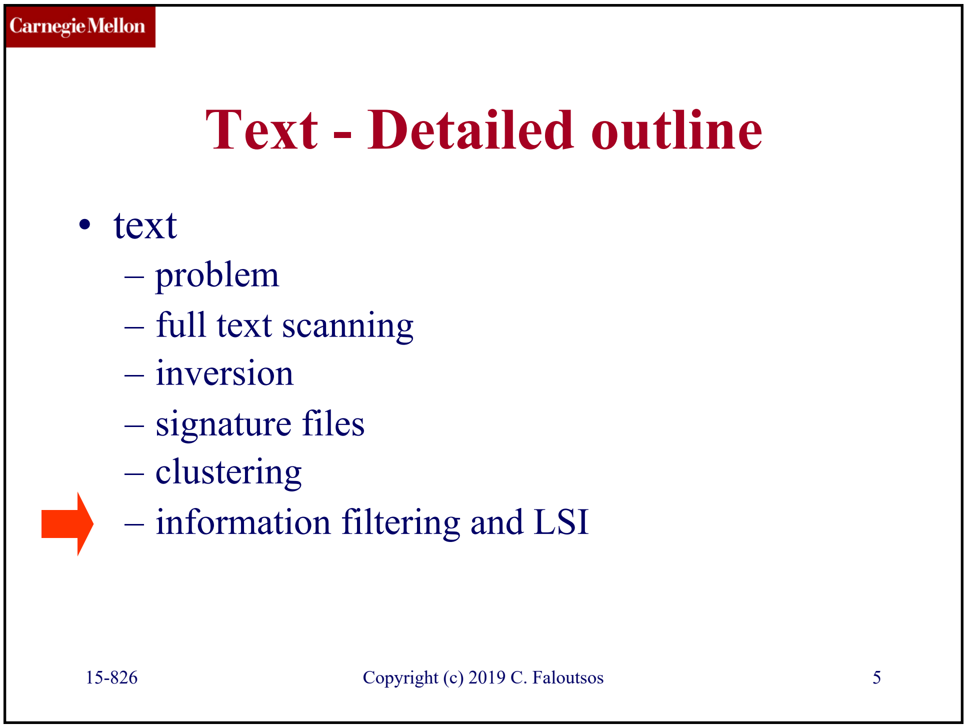


Carnegie Mellon

Indexing - Detailed outline

- primary key indexing
- secondary key / multi-key indexing
- spatial access methods
- fractals
- ➡ text
- SVD: a powerful tool
- multimedia
- ...

15-826 Copyright (c) 2019 C. Faloutsos 4

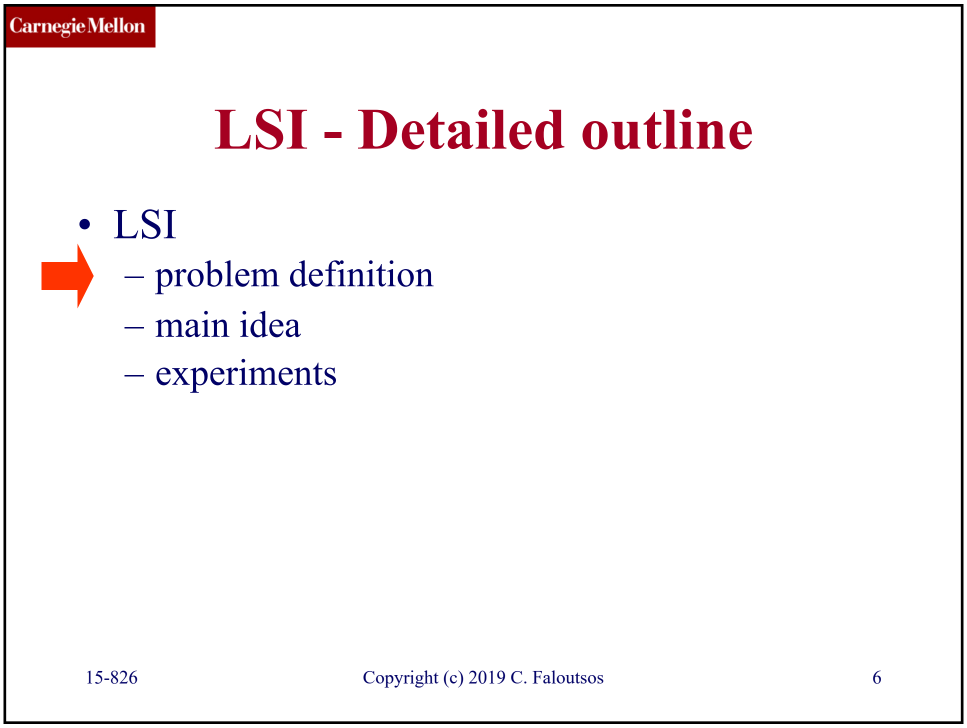


Carnegie Mellon

Text - Detailed outline

- text
  - problem
  - full text scanning
  - inversion
  - signature files
  - clustering
  - ➡ information filtering and LSI

15-826 5

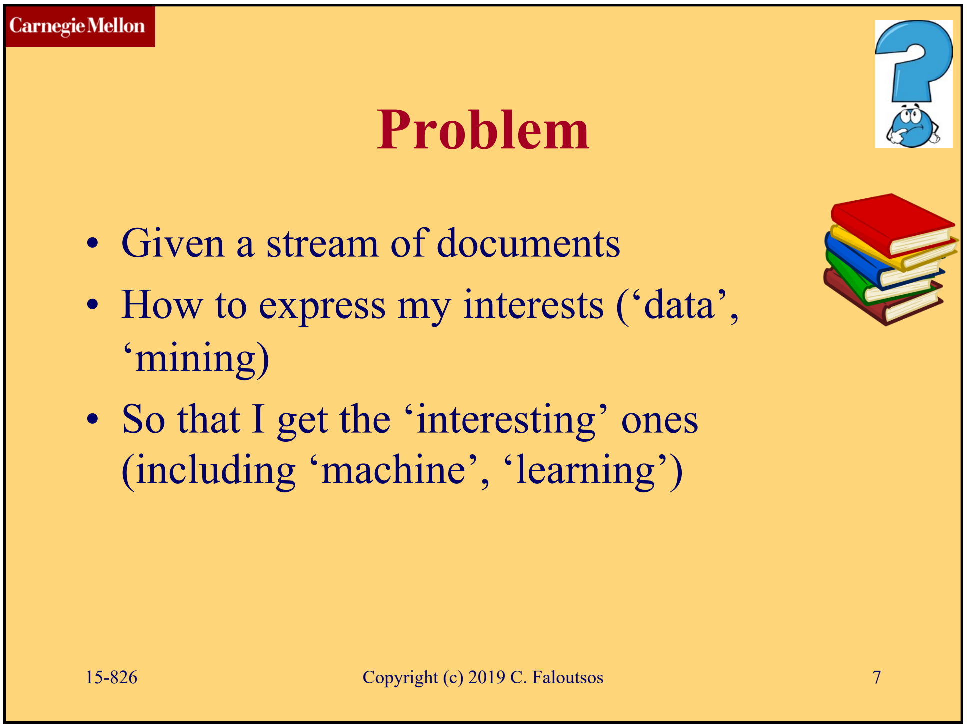


Carnegie Mellon

LSI - Detailed outline

- LSI
  - ➡ problem definition
  - main idea
  - experiments

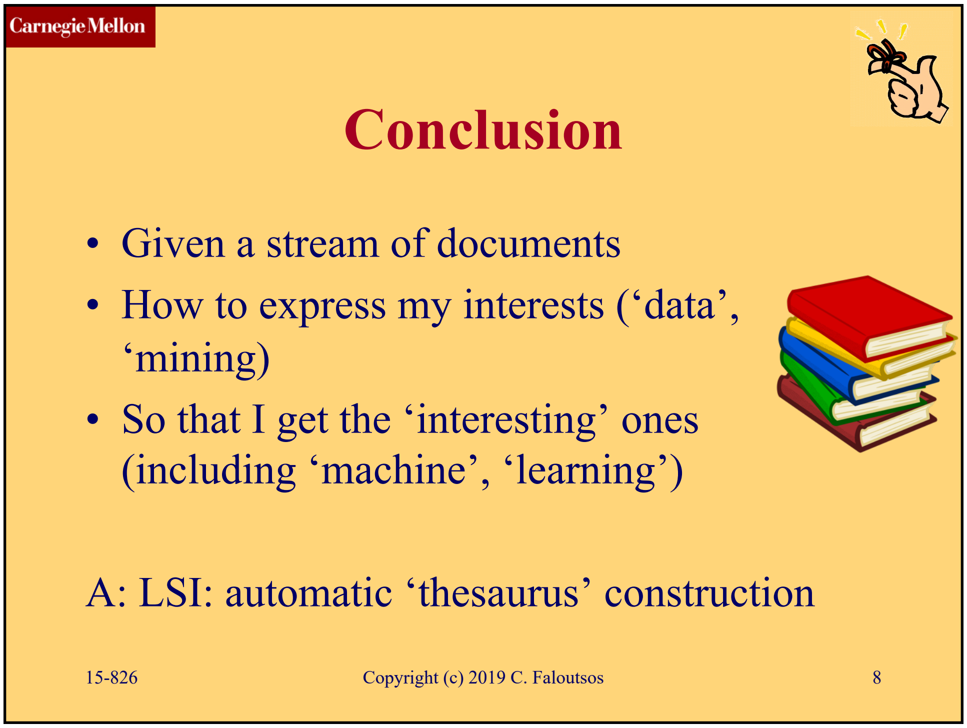
15-826 6



Carnegie Mellon

Problem

- Given a stream of documents
- How to express my interests (‘data’, ‘mining)
- So that I get the ‘interesting’ ones (including ‘machine’, ‘learning’)

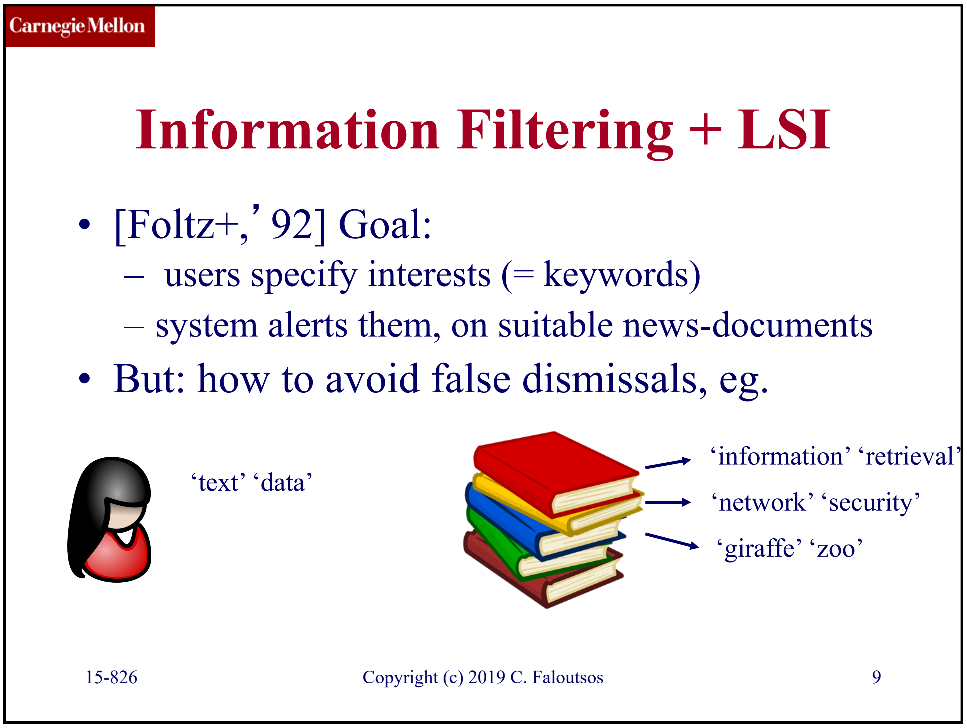


Carnegie Mellon

Conclusion

- Given a stream of documents
- How to express my interests (‘data’, ‘mining)
- So that I get the ‘interesting’ ones (including ‘machine’, ‘learning’)

A: LSI: automatic ‘thesaurus’ construction

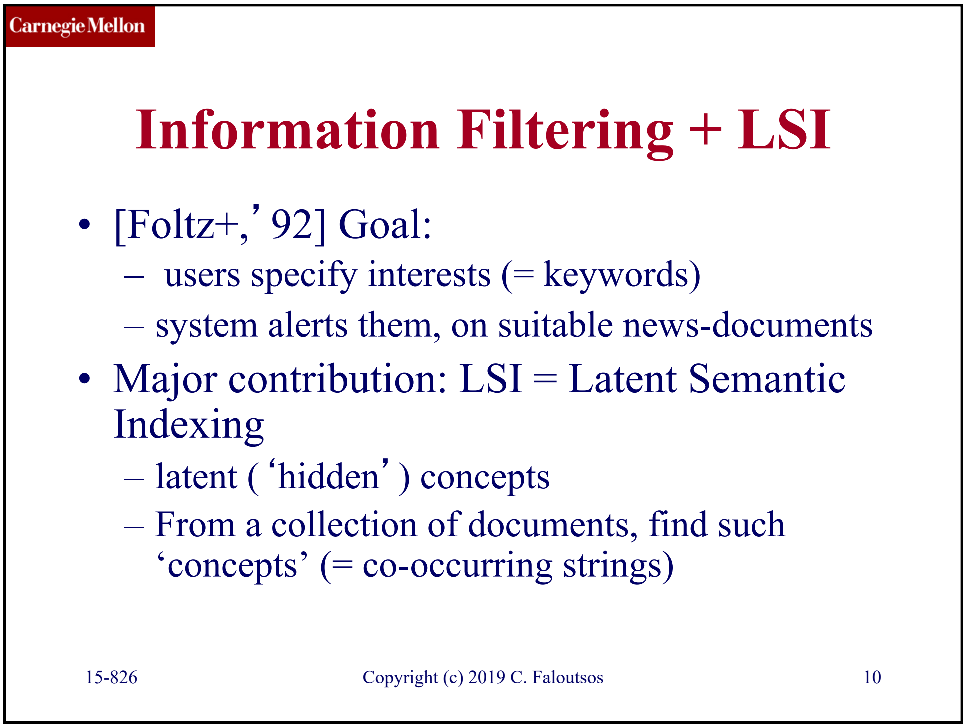


Carnegie Mellon

Information Filtering + LSI

- [Foltz+,’ 92] Goal:
  - users specify interests (= keywords)
  - system alerts them, on suitable news-documents
- But: how to avoid false dismissals, eg.

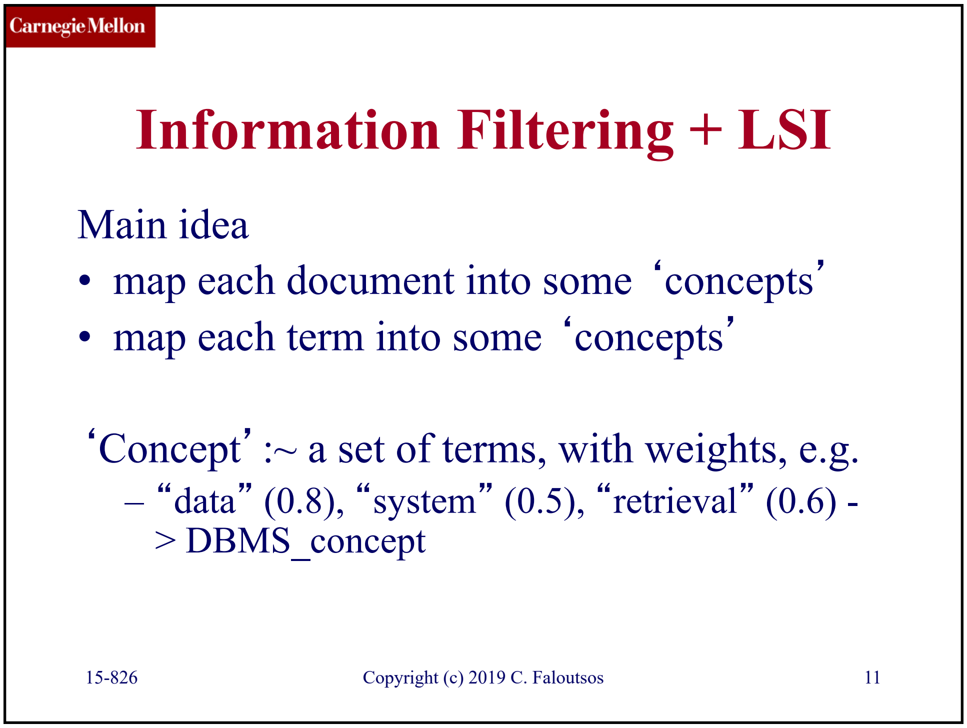
‘text’ ‘data’
‘information’ ‘retrieval’
‘network’ ‘security’
‘giraffe’ ‘zoo’



Carnegie Mellon

Information Filtering + LSI

- [Foltz+,’ 92] Goal:
  - users specify interests (= keywords)
  - system alerts them, on suitable news-documents
- Major contribution: LSI = Latent Semantic Indexing
  - latent (‘hidden’) concepts
  - From a collection of documents, find such ‘concepts’ (= co-occurring strings)



Carnegie Mellon

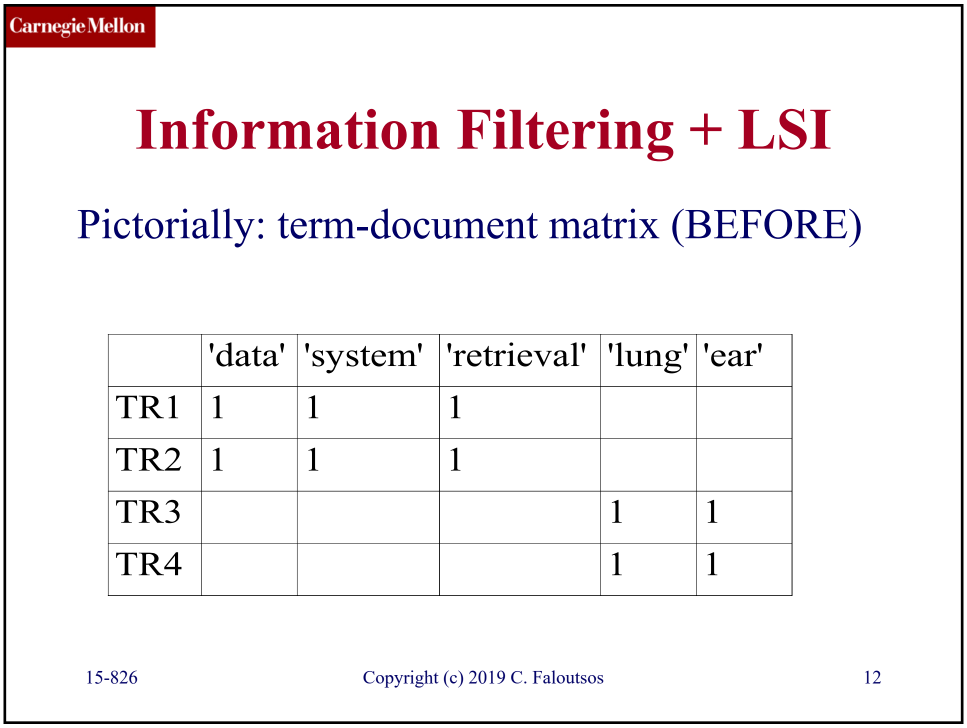
Information Filtering + LSI

Main idea

- map each document into some ‘concepts’
- map each term into some ‘concepts’

‘Concept’ :~ a set of terms, with weights, e.g.

- “data” (0.8), “system” (0.5), “retrieval” (0.6) -> DBMS_concept

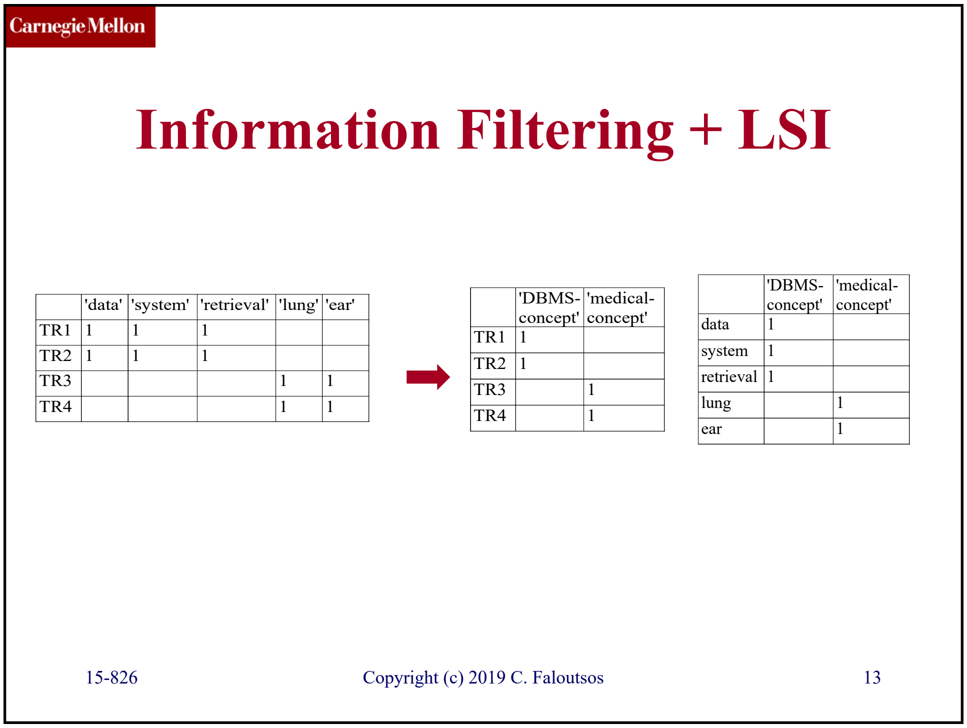


Carnegie Mellon

Information Filtering + LSI

Pictorially: term-document matrix (BEFORE)

	'data'	'system'	'retrieval'	'lung'	'ear'
TR1	1	1	1		
TR2	1	1	1		
TR3				1	1
TR4				1	1



Carnegie Mellon

Information Filtering + LSI

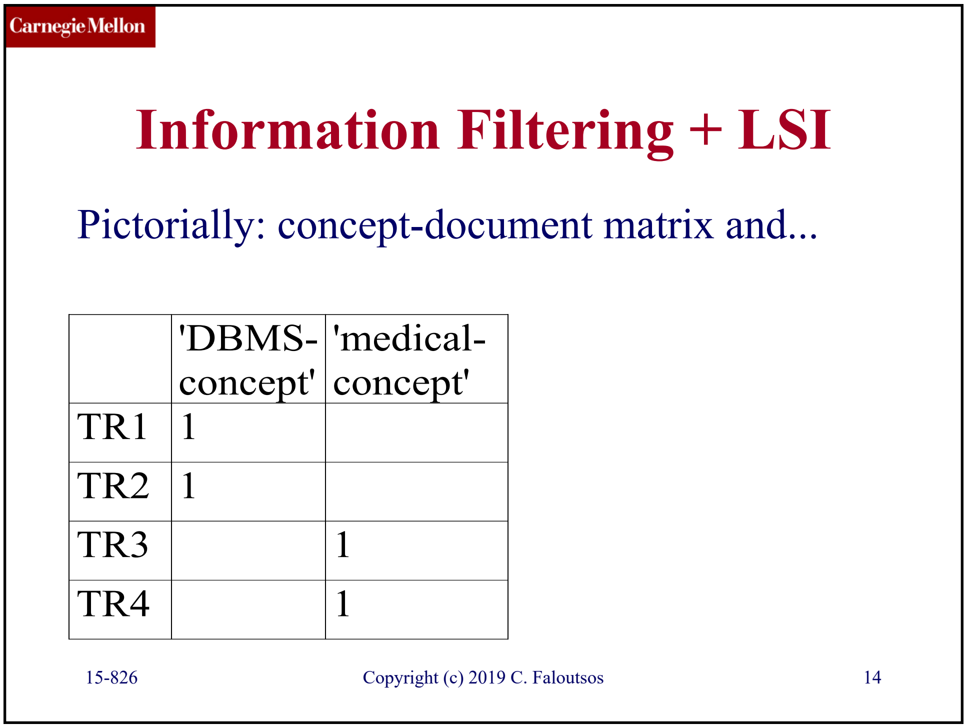
	'data'	'system'	'retrieval'	'lung'	'ear'
TR1	1	1	1		
TR2	1	1	1		
TR3				1	1
TR4				1	1

➡

	'DBMS-concept'	'medical-concept'
TR1	1	
TR2	1	
TR3		1
TR4		1

	'DBMS-concept'	'medical-concept'
data	1	
system	1	
retrieval	1	
lung		1
ear		1

15-826 Copyright (c) 2019 C. Faloutsos 13



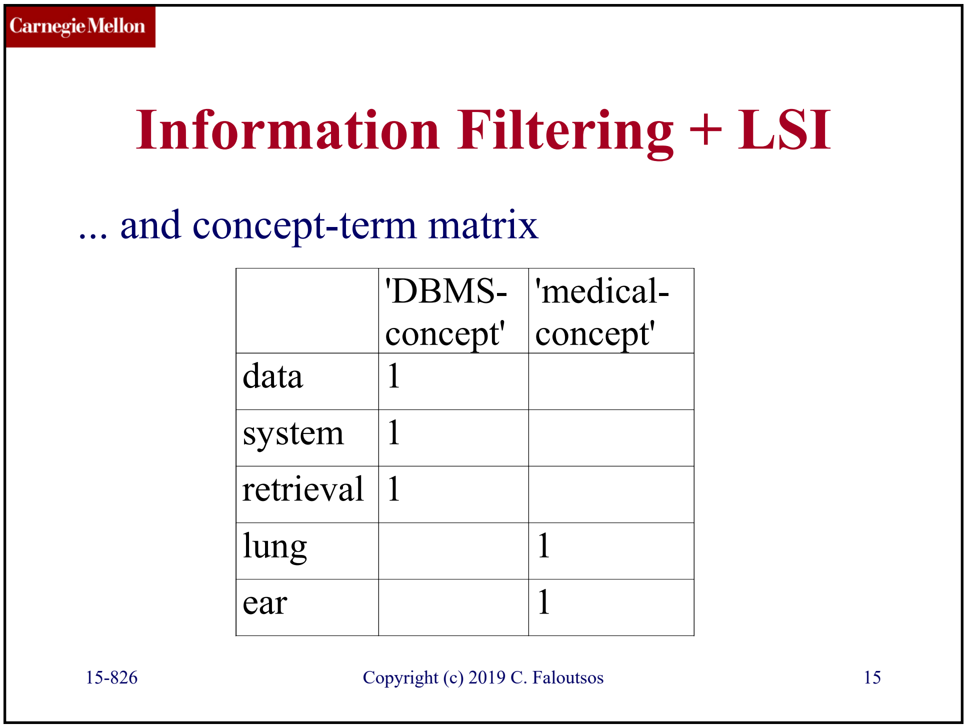
Carnegie Mellon

Information Filtering + LSI

Pictorially: concept-document matrix and...

	'DBMS-concept'	'medical-concept'
TR1	1	
TR2	1	
TR3		1
TR4		1

15-826 Copyright (c) 2019 C. Faloutsos 14

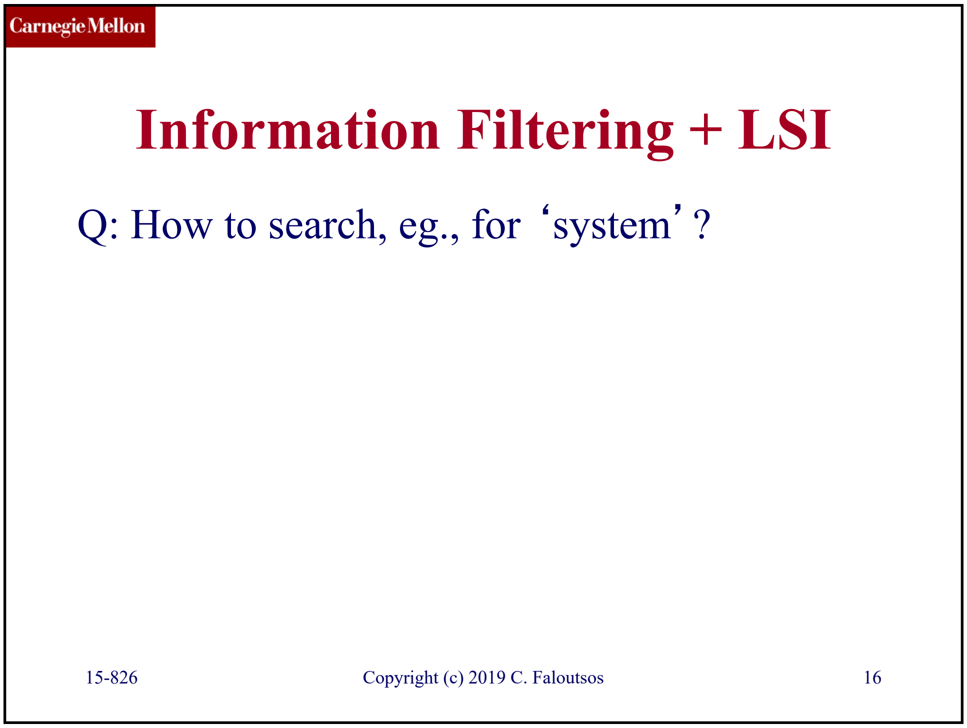


Carnegie Mellon

Information Filtering + LSI

... and concept-term matrix

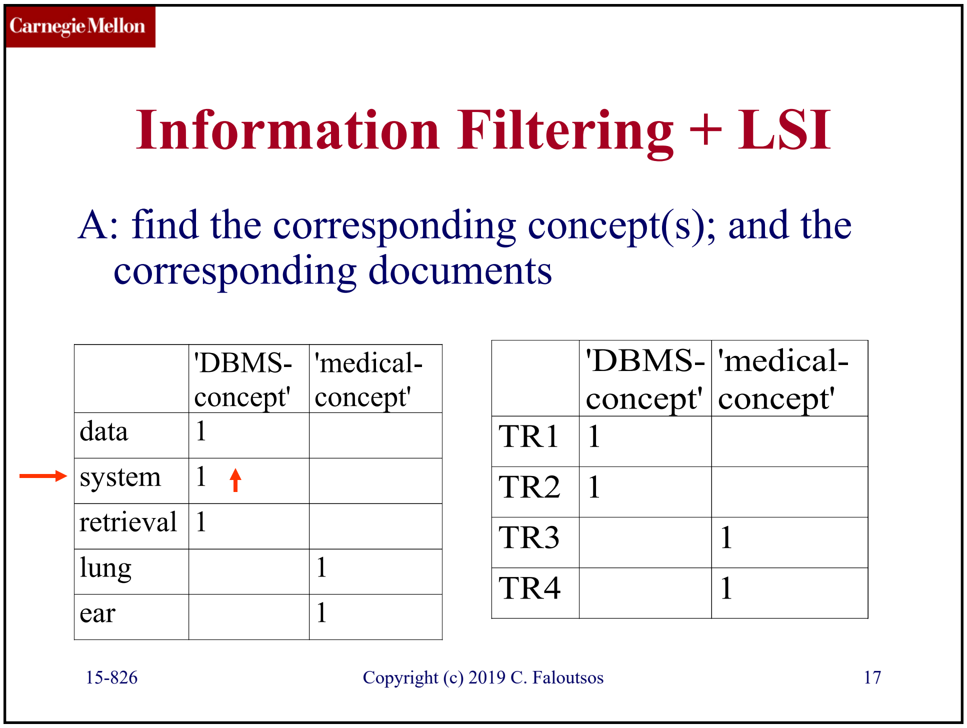
	'DBMS-concept'	'medical-concept'
data	1	
system	1	
retrieval	1	
lung		1
ear		1



Carnegie Mellon

Information Filtering + LSI

Q: How to search, eg., for ‘system’?



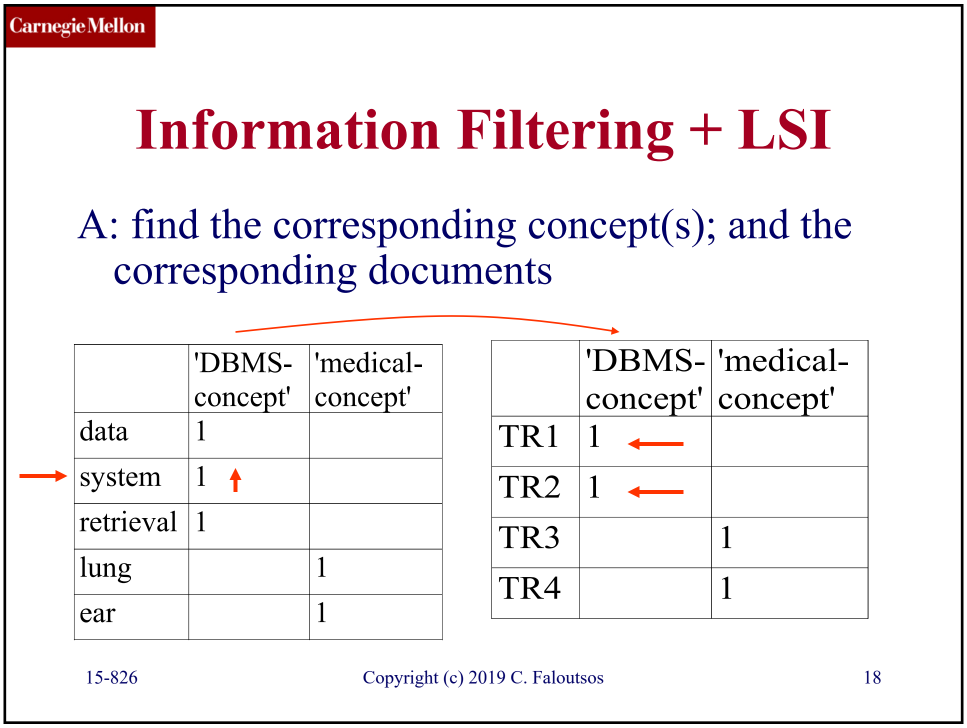
Carnegie Mellon

Information Filtering + LSI

A: find the corresponding concept(s); and the corresponding documents

	'DBMS-concept'	'medical-concept'
data	1	
system	1 ↑	
retrieval	1	
lung		1
ear		1

	'DBMS-concept'	'medical-concept'
TR1	1	
TR2	1	
TR3		1
TR4		1



Carnegie Mellon

Information Filtering + LSI

A: find the corresponding concept(s); and the corresponding documents

	'DBMS-concept'	'medical-concept'
data	1	
system	1 ↑	
retrieval	1	
lung		1
ear		1

	'DBMS-concept'	'medical-concept'
TR1	1 ←	
TR2	1 ←	
TR3		1
TR4		1

Information Filtering + LSI

Thus it works like an (automatically constructed) thesaurus:

we may retrieve documents that DON’T have the term ‘system’, but they contain almost everything else (‘data’, ‘retrieval’)

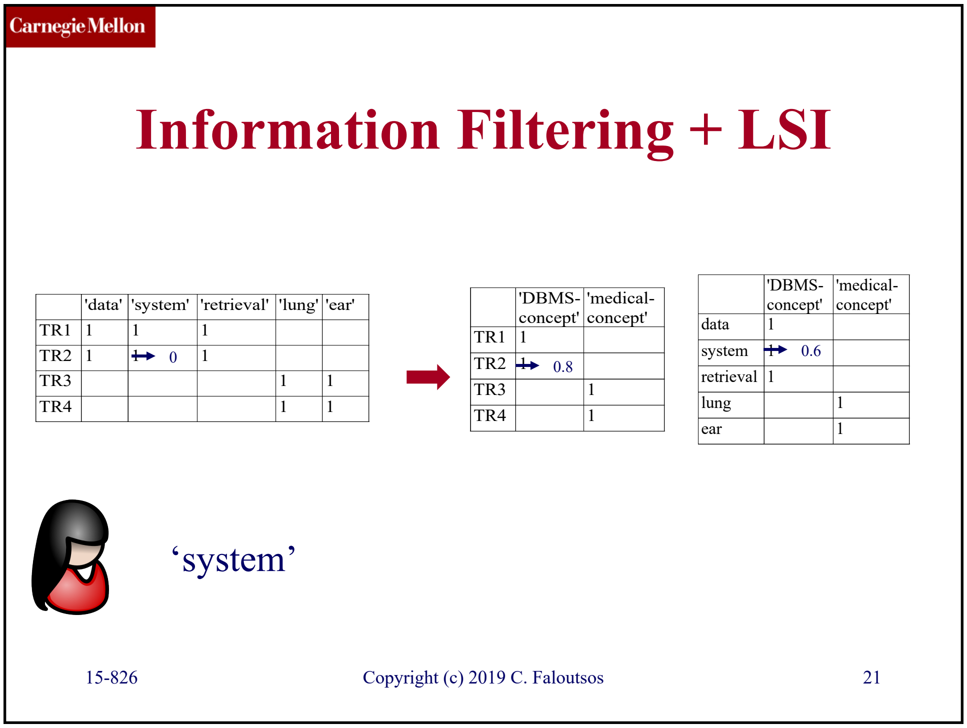
Information Filtering + LSI

	'data'	'system'	'retrieval'	'lung'	'ear'
TR1	1	1	1		
TR2	1	~~1~~ → 0	1		
TR3				1	1
TR4				1	1

→

	'DBMS-concept'	'medical-concept'
TR1	1	
TR2	~~1~~ → 0.8	
TR3		1
TR4		1

	'DBMS-concept'	'medical-concept'
data	1	
system	~~1~~ → 0.6	
retrieval	1	
lung		1
ear		1



Carnegie Mellon

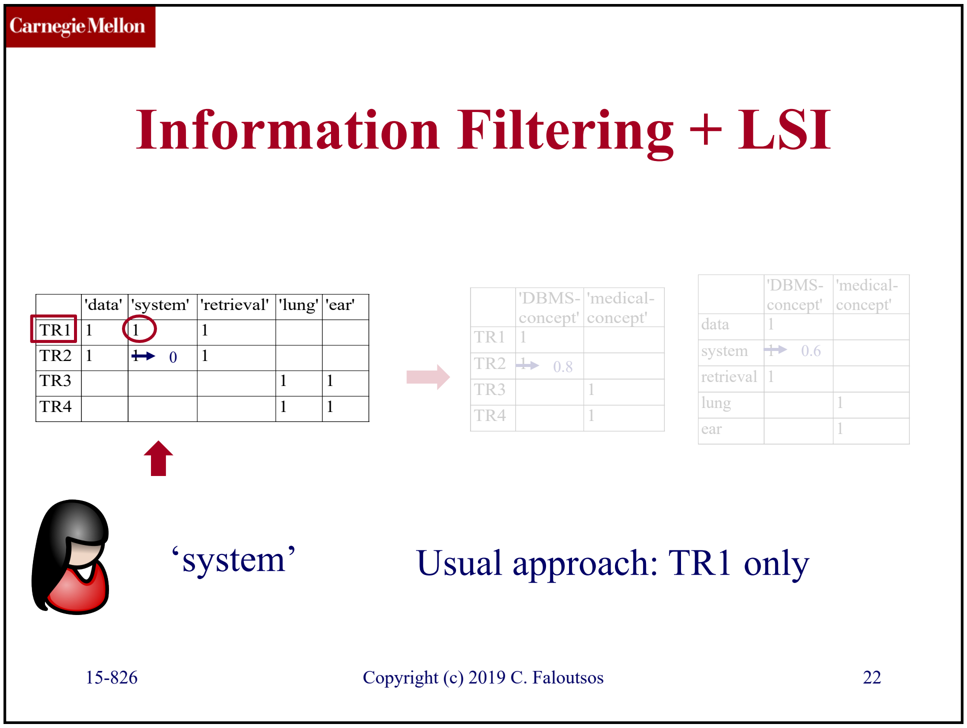
Information Filtering + LSI

	'data'	'system'	'retrieval'	'lung'	'ear'
TR1	1	1	1		
TR2	1	1 → 0	1		
TR3				1	1
TR4				1	1

	'DBMS-concept'	'medical-concept'
TR1	1	
TR2	1 → 0.8	
TR3		1
TR4		1

	'DBMS-concept'	'medical-concept'
data	1	
system	1 → 0.6	
retrieval	1	
lung		1
ear		1

‘system’



Carnegie Mellon

Information Filtering + LSI

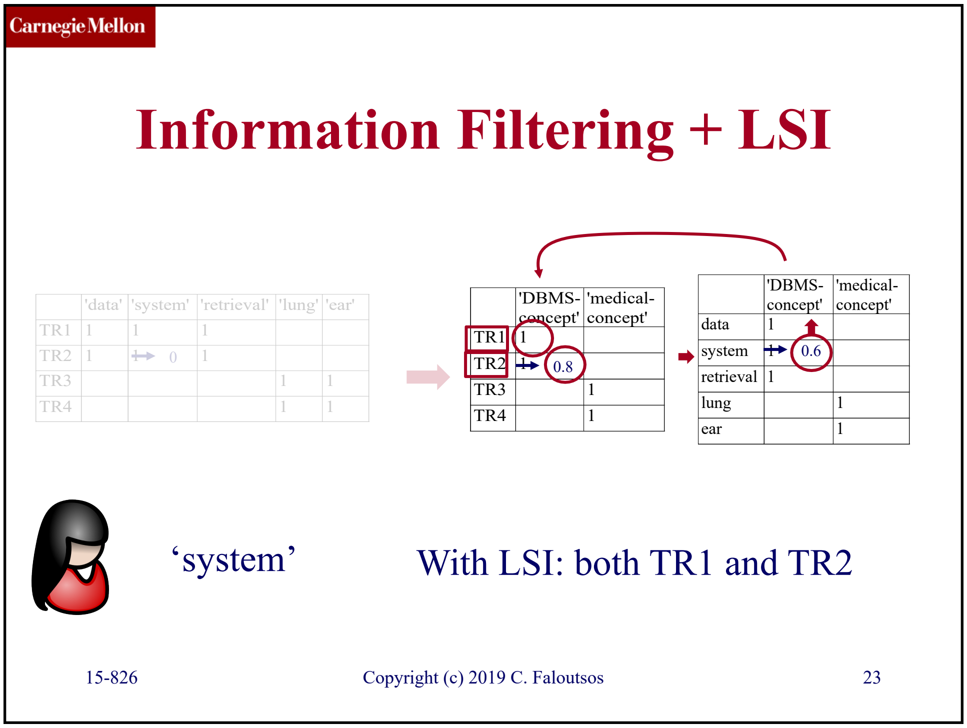
	'data'	'system'	'retrieval'	'lung'	'ear'
TR1	1	1	1		
TR2	1	1 → 0	1		
TR3				1	1
TR4				1	1

	'DBMS-concept'	'medical-concept'
TR1	1	
TR2	1 → 0.8	
TR3		1
TR4		1

	'DBMS-concept'	'medical-concept'
data	1	
system	1 → 0.6	
retrieval	1	
lung		1
ear		1

‘system’

Usual approach: TR1 only



Carnegie Mellon

Information Filtering + LSI

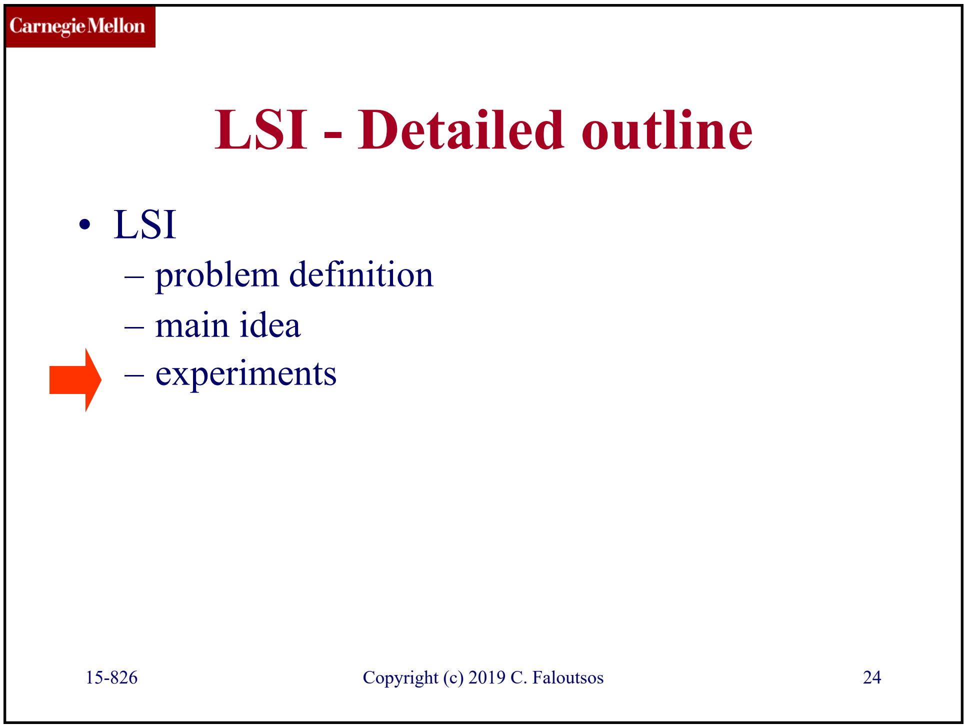
	'data'	'system'	'retrieval'	'lung'	'ear'
TR1	1	1	1		
TR2	1	1 → 0	1		
TR3				1	1
TR4				1	1

	'DBMS-concept'	'medical-concept'
TR1	1	
TR2	1 → 0.8	
TR3		1
TR4		1

	'DBMS-concept'	'medical-concept'
data	1	
system	1 → 0.6	
retrieval	1	
lung		1
ear		1

‘system’

With LSI: both TR1 and TR2



Carnegie Mellon

LSI - Detailed outline

- LSI
  - problem definition
  - main idea
  - experiments

Carnegie Mellon

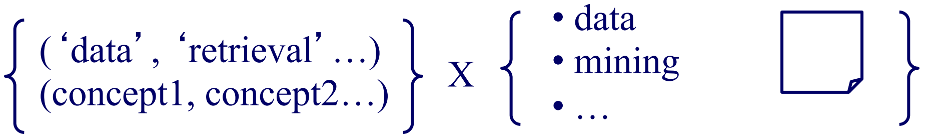
LSI - Experiments

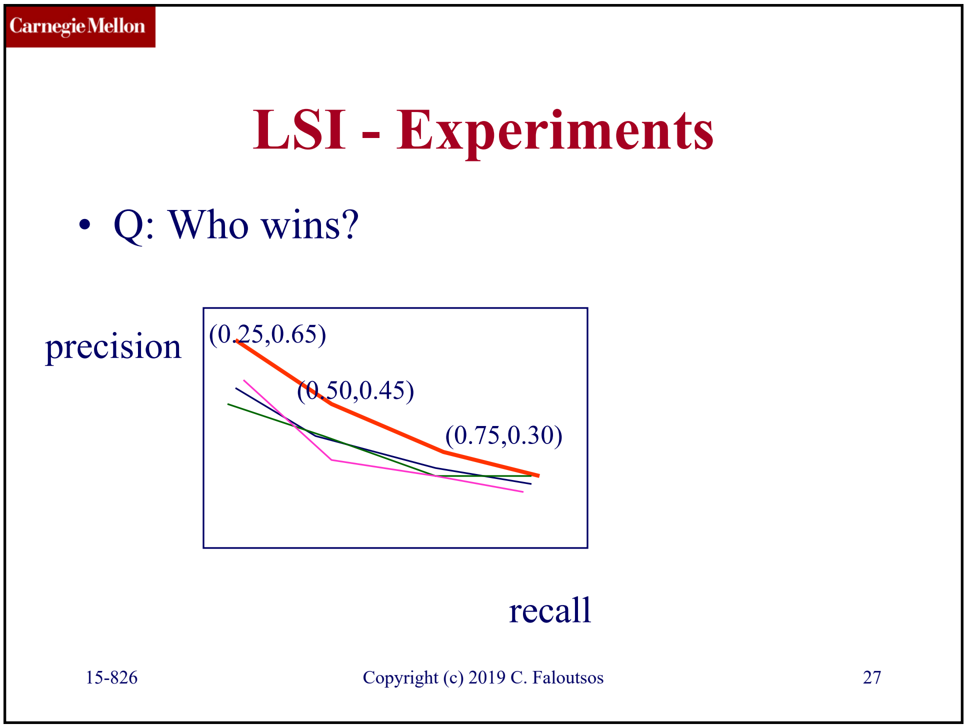
- 150 Tech Memos (TM) / month
- 34 users submitted ‘profiles’ (6-66 words per profile)
- 100-300 concepts

Carnegie Mellon

LSI - Experiments

- four methods, cross-product of:
  - vector-space or LSI, for similarity scoring
  - keywords or document-sample, for profile specification
- measured: precision/recall



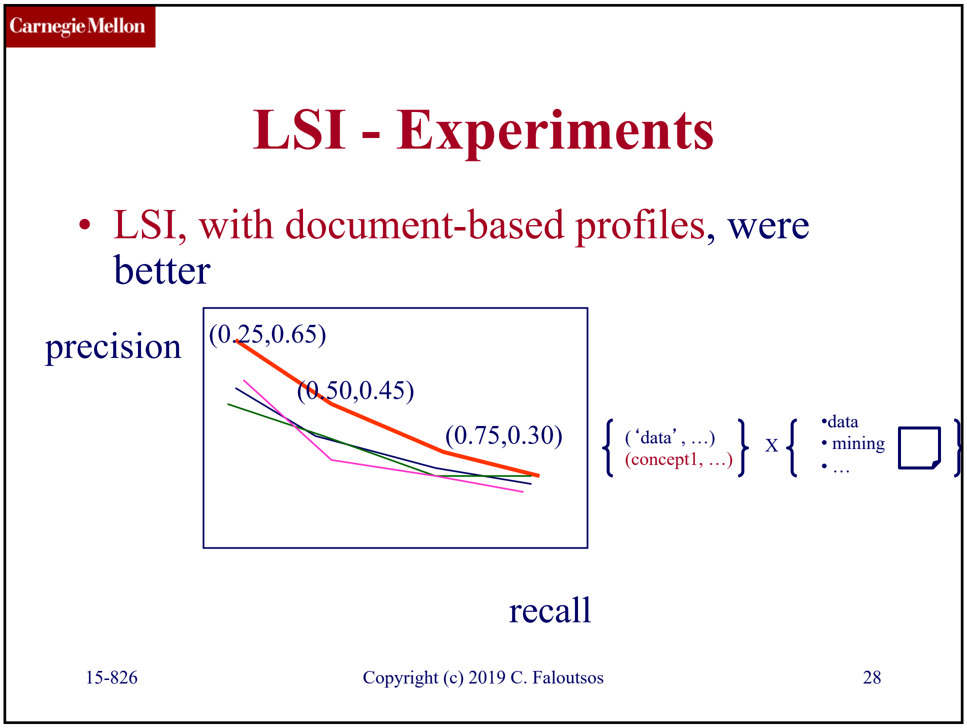


Carnegie Mellon

LSI - Experiments

- Q: Who wins?

15-826 Copyright (c) 2019 C. Faloutsos 27

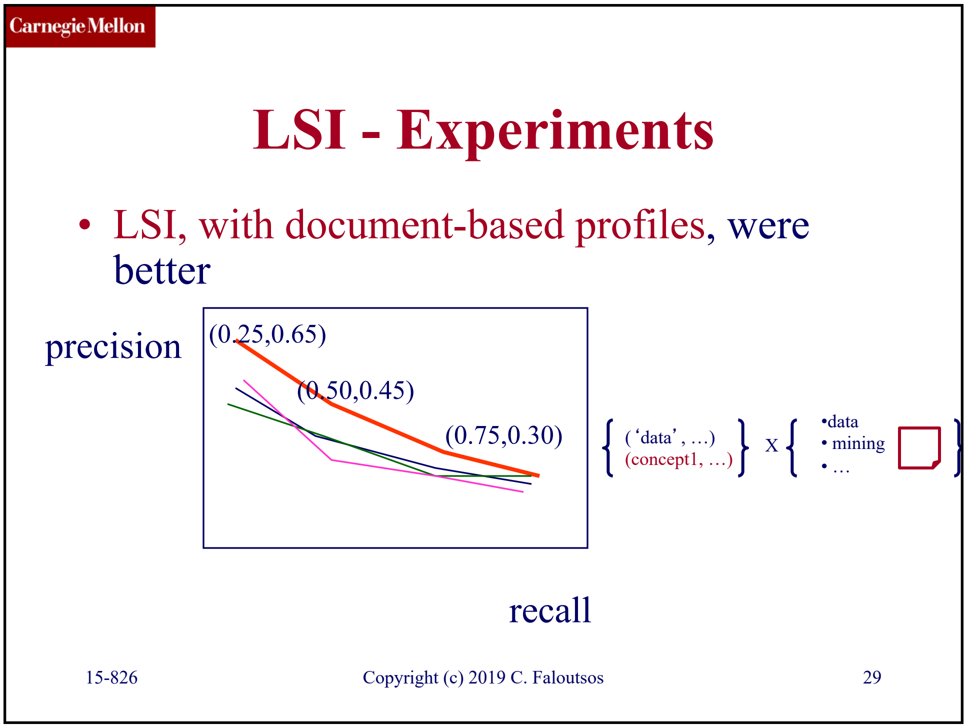


Carnegie Mellon

LSI - Experiments

- LSI, with document-based profiles, were better

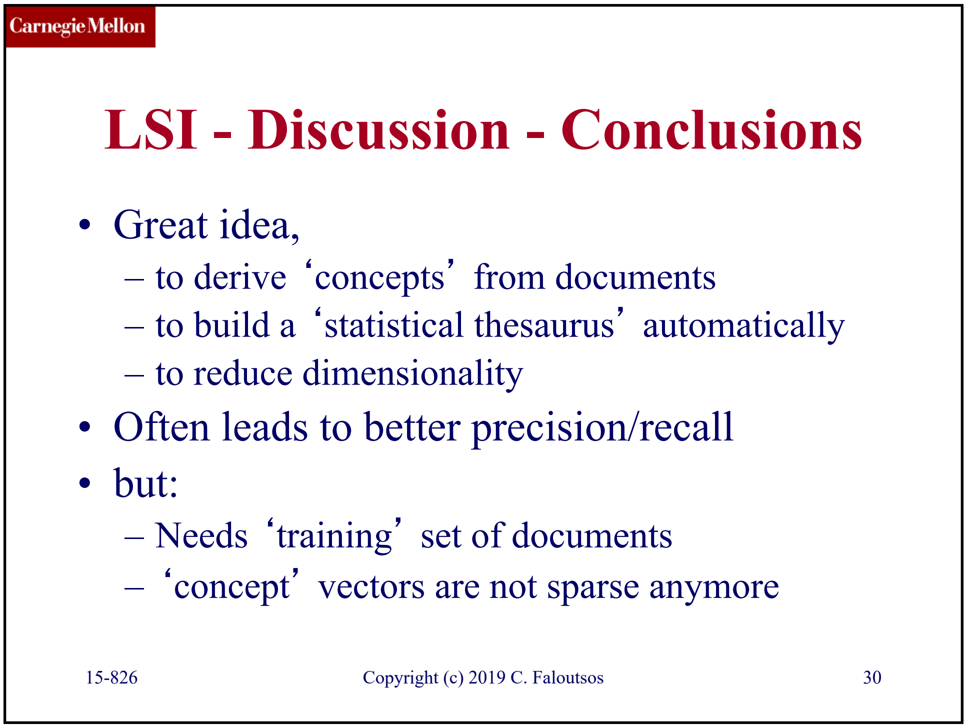
15-826 Copyright (c) 2019 C. Faloutsos 28



Carnegie Mellon

LSI - Experiments

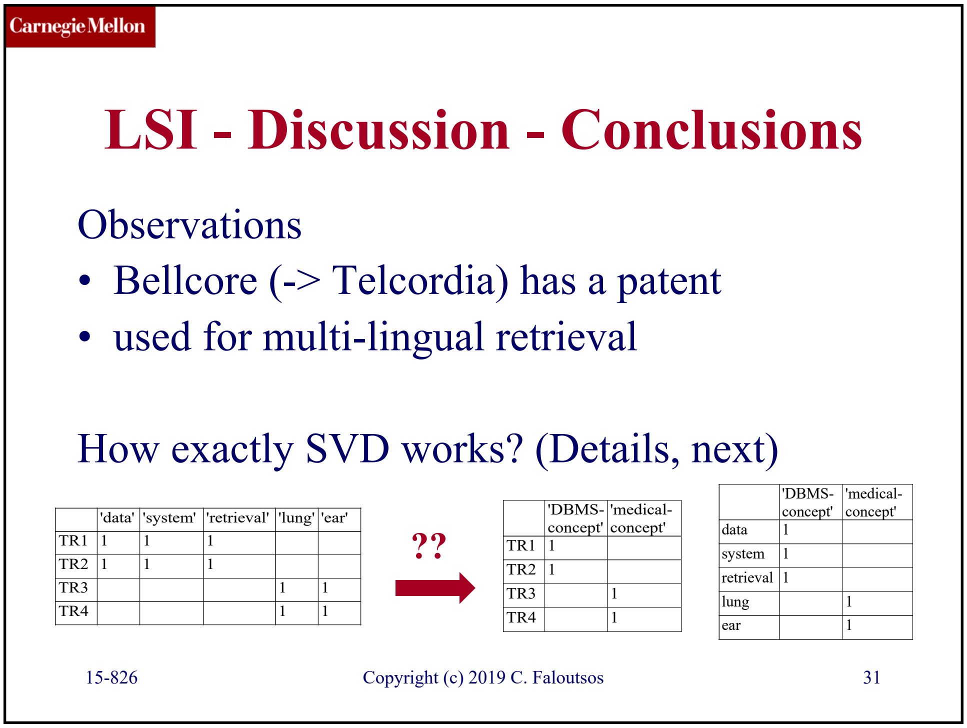
- LSI, with document-based profiles, were better



Carnegie Mellon

LSI - Discussion - Conclusions

- Great idea,
  - to derive 'concepts' from documents
  - to build a 'statistical thesaurus' automatically
  - to reduce dimensionality
- Often leads to better precision/recall
- but:
  - Needs 'training' set of documents
  - 'concept' vectors are not sparse anymore



Carnegie Mellon

LSI - Discussion - Conclusions

Observations

- Bellcore (-> Telcordia) has a patent
- used for multi-lingual retrieval

How exactly SVD works? (Details, next)

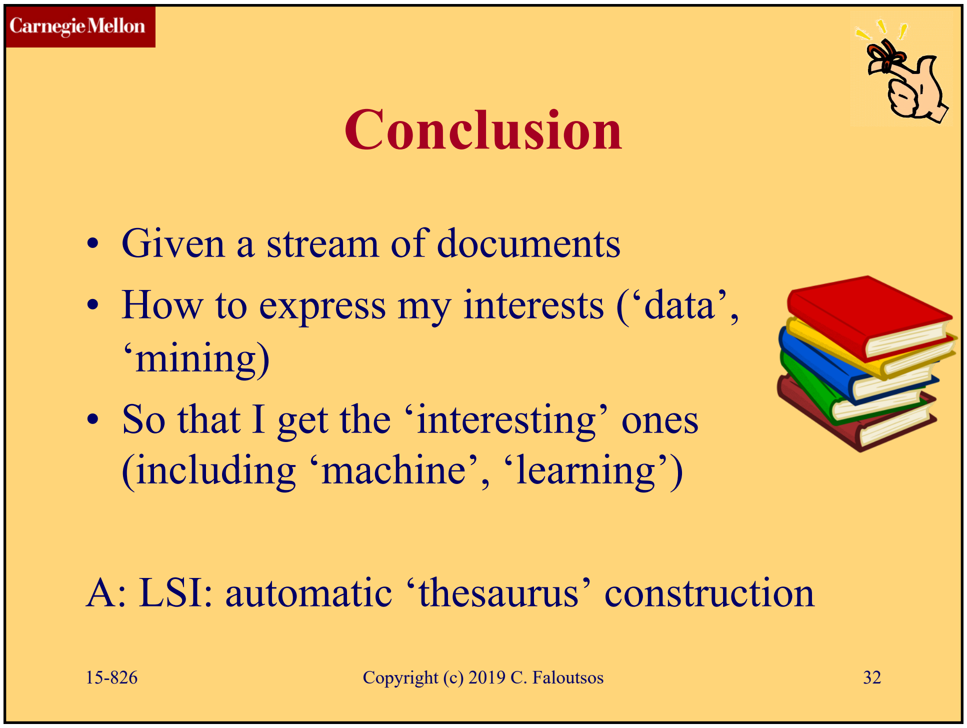
	'data'	'system'	'retrieval'	'lung'	'ear'
TR1	1	1	1		
TR2	1	1	1		
TR3				1	1
TR4				1	1

??

	'DBMS-concept'	'medical-concept'
TR1	1	
TR2	1	
TR3		1
TR4		1

	'DBMS-concept'	'medical-concept'
data	1	
system	1	
retrieval	1	
lung		1
ear		1

15-826 Copyright (c) 2019 C. Faloutsos 31



Carnegie Mellon

Conclusion

- Given a stream of documents
- How to express my interests (‘data’, ‘mining)
- So that I get the ‘interesting’ ones (including ‘machine’, ‘learning’)

A: LSI: automatic ‘thesaurus’ construction

15-826 Copyright (c) 2019 C. Faloutsos 32