
 CMU SCS

15-826: Multimedia Databases and Data Mining

Text - part I
C. Faloutsos


 CMU SCS

Outline

Goal: 'Find similar / interesting things'

- Intro to DB
- ➔ • Indexing - similarity search
- Data Mining

15-826 Copyright: C. Faloutsos (2007) 2

 CMU SCS

Indexing - Detailed outline

- primary key indexing
- secondary key / multi-key indexing
- spatial access methods
- fractals
- ➔ • text
- multimedia
- ...

15-826 Copyright: C. Faloutsos (2007) 3

CMU SCS

Text - Detailed outline

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

15-826 Copyright: C. Faloutsos (2007) 4

CMU SCS

Problem - Motivation

- Eg., find documents containing “*data*”, “*retrieval*”
- Applications:

15-826 Copyright: C. Faloutsos (2007) 5

CMU SCS

Problem - Motivation

- Eg., find documents containing “*data*”, “*retrieval*”
- Applications:
 - Web
 - law + patent offices
 - digital libraries
 - information filtering

15-826 Copyright: C. Faloutsos (2007) 6

CMU SCS

Problem - Motivation

- Types of queries:
 - boolean ('data' AND 'retrieval' AND NOT ...)

15-826 Copyright: C. Faloutsos (2007) 7

CMU SCS

Problem - Motivation

- Types of queries:
 - boolean ('data' AND 'retrieval' AND NOT ...)
 - additional features ('data' ADJACENT 'retrieval')
 - keyword queries ('data', 'retrieval')
- How to search a large collection of documents?

15-826 Copyright: C. Faloutsos (2007) 8

CMU SCS

Full-text scanning

- Build a FSA; scan

15-826 Copyright: C. Faloutsos (2007) 9

CMU SCS

Full-text scanning

- for single term:
 - (naive: $O(N*M)$)

ABRACADABRA text

CAB pattern

15-826 Copyright: C. Faloutsos (2007) 10

CMU SCS

Full-text scanning

- for single term:
 - (naive: $O(N*M)$)
 - Knuth Morris and Pratt ('77)
 - build a small FSA; visit every text letter once only, by carefully shifting more than one step

ABRACADABRA text

CAB pattern

15-826 Copyright: C. Faloutsos (2007) 11

CMU SCS

Full-text scanning

ABRACADABRA text

CAB pattern

15-826 Copyright: C. Faloutsos (2007) 12

CMU SCS

Full-text scanning

- for single term:
 - (naive: $O(N * M)$)
 - Knuth Morris and Pratt ('77)
 - Boyer and Moore ('77)
 - preprocess pattern; start from **right to left & skip!**

ABRACADABRA text

CAB pattern

15-826 Copyright: C. Faloutsos (2007) 13

CMU SCS

Full-text scanning

ABRACADABRA text

CAB pattern

| |

| |

| |

| |

| |

15-826 Copyright: C. Faloutsos (2007) 14

CMU SCS

Full-text scanning

ABRACADABRA text

OMINOUS pattern

|

OMINOUS

Boyer+Moore: fastest, in practice
 Sunday ('90): some improvements

15-826 Copyright: C. Faloutsos (2007) 15

CMU SCS

Full-text scanning

- For multiple terms (w/o “don’t care” characters): Aho+Corasic ('75)
 - again, build a simplified FSA in $O(M)$ time
- Probabilistic algorithms: ‘fingerprints’ (Karp + Rabin '87)
- approximate match: ‘agrep’ [Wu+Manber, Baeza-Yates+, '92]

15-826 Copyright: C. Faloutsos (2007) 16

CMU SCS

Full-text scanning

- Approximate matching - **string editing** distance:
 $d(\text{'survey'}, \text{'surgery'}) = 2$
 = min # of insertions, deletions, substitutions to transform the first string into the second

SURVEY
 SURGERY

15-826 Copyright: C. Faloutsos (2007) 17

CMU SCS

Full-text scanning

- **string editing** distance - how to compute?
- A:

15-826 Copyright: C. Faloutsos (2007) 18

CMU SCS

Full-text scanning

- **string editing** distance - how to compute?
- A: dynamic programming
 - cost(i, j) = cost to match prefix of length i of first string s with prefix of length j of second string t

15-826 Copyright: C. Faloutsos (2007) 19

CMU SCS

Full-text scanning

if $s[i] = t[j]$ then
 cost(i, j) = cost($i-1, j-1$)
 else
 cost(i, j) = min (

- 1 + cost($i, j-1$) // deletion
- 1 + cost($i-1, j-1$) // substitution
- 1 + cost($i-1, j$) // insertion

)

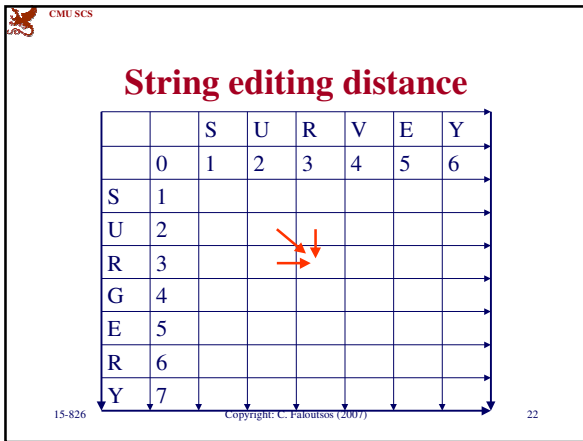
15-826 Copyright: C. Faloutsos (2007) 20

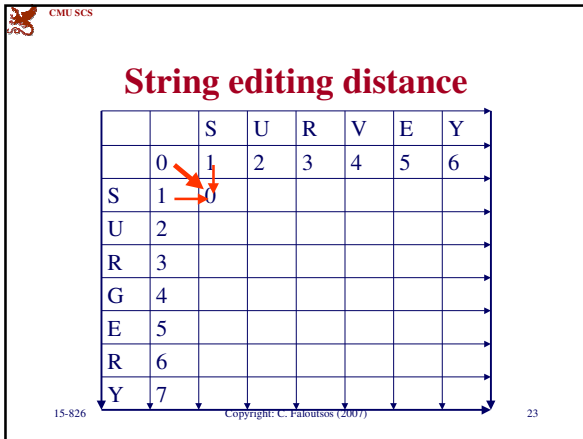
CMU SCS

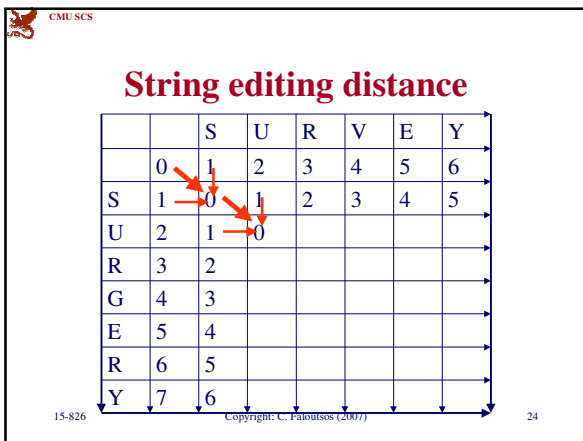
String editing distance

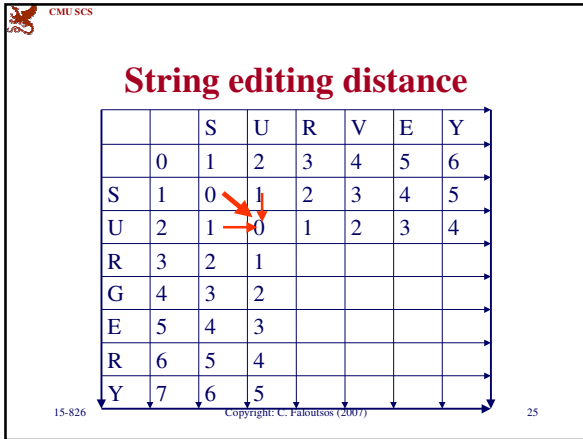
	ϕ	S	U	R	V	E	Y
ϕ	0	1	2	3	4	5	6
S	1						
U	2						
R	3						
G	4						
E	5						
R	6						
Y	7						

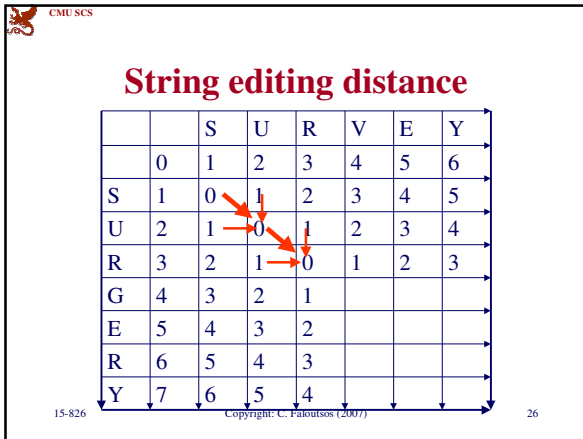
15-826 Copyright: C. Faloutsos (2007) 21













CMU SCS

String editing distance

		S	U	R	V	E	Y
	0	1	2	3	4	5	6
S	1	0	1	2	3	4	5
U	2	1	0	1	2	3	4
R	3	2	1	0	1	2	3
G	4	3	2	1	1	2	3
E	5	4	3	2	2	1	2
R	6	5	4	3	3	2	
Y	7	6	5	4	4	3	

15-826 Copyright: C. Faloutsos (2007) 28

CMU SCS

String editing distance

		S	U	R	V	E	Y
	0	1	2	3	4	5	6
S	1	0	1	2	3	4	5
U	2	1	0	1	2	3	4
R	3	2	1	0	1	2	3
G	4	3	2	1	1	2	3
E	5	4	3	2	2	1	2
R	6	5	4	3	3	2	2
Y	7	6	5	4	4	3	

15-826 Copyright: C. Faloutsos (2007) 29

CMU SCS

String editing distance

		S	U	R	V	E	Y
	0	1	2	3	4	5	6
S	1	0	1	2	3	4	5
U	2	1	0	1	2	3	4
R	3	2	1	0	1	2	3
G	4	3	2	1	1	2	3
E	5	4	3	2	2	1	2
R	6	5	4	3	3	2	2
Y	7	6	5	4	4	3	2

15-826 Copyright: C. Faloutsos (2007) 30

CMU SCS

String editing distance

		S	U	R	V	E	Y
	0	1	2	3	4	5	6
S	1	0	1	2	3	4	5
U	2	1	0	1	2	3	4
R	3	2	1	0	1	2	3
G	4	3	2	1	1	2	3
E	5	4	3	2	2	1	2
R	6	5	4	3	3	2	2
Y	7	6	5	4	4	3	2

15-826 Copyright: C. Faloutsos (2007) 31

CMU SCS

String editing distance

		S	U	R	V	E	Y
	0	1	2	3	4	5	6
S	1	0	1	2	3	4	5
U	2	1	0	1	2	3	4
R	3	2	1	0	1	2	3
G	4	3	2	1	1	2	3
E	5	4	3	2	2	1	2
R	6	5	4	3	3	2	2
Y	7	6	5	4	4	3	2

15-826 Copyright: C. Faloutsos (2007) 32

CMU SCS

String editing distance

		S	U	R	V	E	Y
	0	1	2	3	4	5	6
S	1	0	1	2	3	4	5
U	2	1	0	1	2	3	4
R	3	2	1	0	1	2	3
G	4	3	2	1	1	2	3
E	5	4	3	2	2	1	2
R	6	5	4	3	3	2	2
Y	7	6	5	4	4	3	2

15-826 Copyright: C. Faloutsos (2007) 33

CMU SCS

String editing distance

		S	U	R	V	E	Y
	0	1	2	3	4	5	6
S	1	0	1	2	3	4	5
U	2	1	0	1	2	3	4
R	3	2	1	0	1	2	3
G	4	3	2	1	1	2	3
E	5	4	3	2	2	1	2
R	6	5	4	3	3	2	2
Y	7	6	5	4	4	3	2

15-826 Copyright: C. Faloutsos (2007) 34

CMU SCS

String editing distance

		S	U	R	V	E	Y
	0	1	2	3	4	5	6
S	1	0	1	2	3	4	5
U	2	1	0	1	2	3	4
R	3	2	1	0	1	2	3
G	4	3	2	1	1	2	3
E	5	4	3	2	2	1	2
R	6	5	4	3	3	2	2
Y	7	6	5	4	4	3	2

subst.
del.


15-826 Copyright: C. Faloutsos (2007) 35

CMU SCS

Full-text scanning

Complexity: $O(M*N)$ (when using a matrix to 'memoize' partial results)

15-826 Copyright: C. Faloutsos (2007) 36

 CMU SCS

Full-text scanning

Conclusions:

- Full text scanning needs no space overhead, but is slow for large datasets

15-826 Copyright: C. Faloutsos (2007) 37
