02A – INFORMATION RETRIEVAL

CS 1656 Introduction to Data Science

Alexandros Labrinidis – http://labrinidis.cs.pitt.edu
University of Pittsburgh

What is Information Retrieval?

- Information organized into documents
  - Large number of documents
  - Data in documents is **unstructured**
- Our mission, should we choose to accept it:
  - Locate documents that match a user’s needs
  - How:
    - Keywords
    - Sample documents
- Like finding a needle in a haystack ☺
  - Or worse: a hay-colored needle!
  - *this isn't mission difficult, it's mission impossible!*

Info Retrieval vs Database Systems

- Database Systems
  - Structured data
  - Complex data models
  - Data updates
  - Transactions and concurrency control
  - Exact Answers
  - Sorted results

- Information Retrieval
  - Unstructured data
  - Collection of documents
  - Mostly static
  - Approximate answers
  - Ranking of results

How to retrieve information

- One way:
  - Get keywords from user
  - Scan entire collection of documents
  - Return documents that match
  - Problems?
- **Will not scale** to large document collections
  - E.g., the Web
- **Will not rank** results
  - E.g., too many matches for "Labrinidis"

Classic Information Retrieval

- Collection of documents D_i, where $0 < i < N$

- One or more keywords k_x, where $0 < x < t$

- **Task**:
  - Given keywords from user
  - Identify documents from collection that contain keywords
  - Rank documents in some way, with most relevant documents first

Sample Document

The Cleveland Browns stunned the Pittsburgh Steelers with an epic second-half turnaround to tie the score in the fourth quarter. But Shaun Suisham kicked a 41-yard field goal with no time left to pull out a 30-27 victory Sunday at Heinz Field that seemed assured at halftime.

- Q1: Should we use all words?
- A1: Should remove stopwords (articles and connectives)
  - E.g., the, with, an, to, in, a, at, that, …

- Q2: Should we preprocess any words?
- A2: Should perform stemming
  - Reduce words to common grammatical root, e.g., stunned -> stun

Are all terms equally relevant?

- Imagine two documents:
  - **Document A:**
    - The University of Pittsburgh is located in Pittsburgh.
  - **Document B:**
    - Carnegie Mellon University is located in Pittsburgh.
- Q: Is one of the two documents more “relevant” with respect to a certain keyword?
  - (i.e., expect it higher in the ranked results)
- Q: Which keyword?

Relevance Ranking – Single Keyword

- How relevant is document d_j to keyword k_i ?

- Approach #1 -- Frequency
  - Use the number of occurrences of k_i in d_j (frequency)
  - $f(k_i , d_j)$

- Approach #2 – Term Frequency
  - $tf(k_i , d_j) = 1 + \log_2 f(k_i , d_j)$, if $f(k_i , d_j) > 0$
    $= 0$, otherwise

Term-Document Matrix

- The occurrence of a term k_i in a document d_j establishes a relation between k_i and d_j
- A term-document relation between k_i and d_j can be quantified by the frequency of term k_i in document d_j
- In matrix form, this can be written as:

$$\begin{array}{c|cc} & d_1 & d_2 \\ \hline k_1 & F_{1,1} & F_{1,2} \\ k_2 & F_{2,1} & F_{2,2} \\ k_3 & F_{3,1} & F_{3,2} \end{array}$$

- where $F_{i,j}$ is the frequency of keyword i in document $_j$

Example

- Assume the following four documents:

To do is to be.
To be is to do.
d_1

To be or not to be.
I am what I am.
d_2

I think therefore I am.
Do be do be do.
d_3

Do do do, da da da.
Let it be, let it be.
d_4

[Source: Modern Information Retrieval, 2nd Edition]

#	Term	F i, 1	F i, 2	F i,3	F i, 4	TF i, 1	TF i, 2	TF i,3	TF i, 4
1	to								
2	do								
3	is								
4	be								
5	or								
6	not								
7	I								
8	am								
9	what								
10	think								
11	therefore								
12	da								
13	let								
14	it								
Doc Size (# words)									

#	Term	F i, 1	F i, 2	F i,3	F i, 4	TF i, 1	TF i, 2	TF i,3	TF i, 4
1	to	4	2			3	2		
2	do	2		3	3	2		2.585	2.585
3	is	2				2			
4	be	2	2	2	2	2	2	2	2
5	or		1				1		
6	not		1				1		
7	I		2	2			2	2	
8	am		2	1			2	1	
9	what		1				1		
10	think			1				1	
11	therefore			1				1	
12	da				3				2.585
13	let				2				2
14	it				2				2
Doc Size (# words)		10	11	10	12				

What is result of query with keyword = “to” ?

How to handle multiple keywords?

- Most queries involve more than one keywords.
- **Q**: How can we implement relevance ranking over a collection of documents using multiple keywords?
- **A1** – Simple approach:
  - Compute independent relevance measures
  - Add them up
- **A2** – Better approach:
  - Determine importance (weight) of each keyword
  - Compute independent relevance measures
  - Compute weighted sum

How to determine weights?

- **Idea:**
  keywords that do not appear in many documents should be more important than those that do

- **Def**: Inverse document frequency (IDF_i) for keyword k_i
  - $IDF_i = \log_2 (N / n_i)$
  - where
    - n_i = number of documents where ki appears
    - N total number of documents

Putting it all together

- Term **weight** associated with pair k_i , d_j :

- $$W_{i,\, j} = \overbrace{(1 + \log_2 f_{i,j})}^{\text{TF (ki, dj)}} \text{ x } \overbrace{\log_2 (N / n_i)}^{\text{IDF (ki)}}, \qquad \text{if } f_{i,j} > 0$$
  $$= 0, \qquad \text{otherwise}$$

- Variants for first part:
  - $\{0, 1\}$
  - $f_{i,j}$
  - $1 + \log_2 f_{i,j}$

- Variants for second part:
  - 1
  - $\log_2 (N / n_i)$

#	Term	n i	IDF i	d 1	d 2	d 3	d 4
1	to	2	1				
2	do	3	0.415				
3	is	1	2				
4	be	4	0				
5	or	1	2				
6	not	1	2				
7	I	2	1				
8	am	2	1				
9	what	1	2				
10	think	1	2				
11	therefore	1	2				
12	da	1	2				
13	let	1	2				
14	it	1	2				

#	Term	n i	IDF i	d 1	d 2	d 3	d 4
1	to	2	1	3	2		
2	do	3	0.415	0.830		1.073	1.073
3	is	1	2	4			
4	be	4	0				
5	or	1	2		2		
6	not	1	2		2		
7	I	2	1		2	2	
8	am	2	1		2	1	
9	what	1	2		2		
10	think	1	2			2	
11	therefore	1	2			2	
12	da	1	2				5.170
13	let	1	2				4
14	it	1	2				4

Another Example

- Document #1:

  The University of Pittsburgh is located in Pittsburgh.

- Document #2:

  Carnegie Mellon University is located in Pittsburgh.

- Document #3

  Pittsburgh was voted most livable city. Steelers. Steelers!

- Document #4:

  The Steelers won over the Cleveland Browns.

- Document #5:

  The Pittsburgh Steelers have won 6 Super Bowls.

- Document #6:

  Cleveland is located in Ohio.

For keyword = Pittsburgh

	Doc #1	Doc #2	Doc #3	Doc #4	Doc #5	Doc #6
F(Pittsburgh,j)	2	1	1	0	1	0
TF(Pittsburgh,j)	1+log 2=2	1	1	0	1	0

n (Pittsburgh) = 4

IDF (Pittsburgh) = $\log_2 (6 / 4) = 0.585$

	Doc #1	Doc #2	Doc #3	Doc #4	Doc #5	Doc #6
w(Pittsburgh, j)	1.170	0.585	0.585	0	0.585	0

Handout & Pop Quiz

Understanding Question

- **Question**:
  - What is the IDF for the keyword steelers?
- **Possible Answers**:
  - 2.585
  - 2.322
  - 2
  - 1.585
  - 1

Handout Solutions

Q1	Doc #1	Doc #2	Doc #3	Doc #4	Doc #5	Doc #6
F(Steelers. j)	0	0	2	1	1	0
TF(Steelers. j)	0	0	2	1	1	0

Q2: n (Steelers) = 3

IDF (Steelers) = $\log_2$ (6 / 3) = 1 (answer)

Q3	Doc #1	Doc #2	Doc #3	Doc #4	Doc #5	Doc #6
w(Steelers, j)	0	0	2	1	1	0

Query= Pittsburgh + Steelers

	Doc #1	Doc #2	Doc #3	Doc #4	Doc #5	Doc #6
w(Pittsburgh, j)	1.170	0.585	0.585	0	0.585	0

	Doc #1	Doc #2	Doc #3	Doc #4	Doc #5	Doc #6
w(Steelers, j)	0	0	2	1	1	0

	Doc #1	Doc #2	Doc #3	Doc #4	Doc #5	Doc #6
w(Pittsburgh+Steelers, j)	1.170	0.585	2.585	1	1.585	0

Another Example – Results

Query = Pittsburgh + Steelers

- Document #1: 1.170

  The University of Pittsburgh is located in Pittsburgh.
- Document #2: 0.585

  Carnegie Mellon University is located in Pittsburgh.
- Document #3 2.585

  Pittsburgh was voted most livable city. Steelers. Steelers!
- Document #4: 1.0

  The Steelers won over the Cleveland Browns.
- Document #5: 1.585

  The Pittsburgh Steelers have won 6 Super Bowls.
- Document #6: 0

  Cleveland is located in Ohio.

Another Example – Sorted Results

Query = Pittsburgh + Steelers

- Document #3 2.585

  Pittsburgh was voted most livable city. Steelers. Steelers!

- Document #5: 1.585

  The Pittsburgh Steelers have won 6 Super Bowls

- Document #1: 1.170

  The University of Pittsburgh is located in Pittsburgh.

- Document #4: 1.0

  The Steelers won over the Cleveland Browns.

- Document #2: 0.585

  Carnegie Mellon University is located in Pittsburgh.

- ~~Document #6~~: 0

  Cleveland is located in Ohio.

IDF(Steelers) = 1

IDF(Pittsburgh) = 0.585

HOW TO MAKE IR SCALE?

Scaling to large collections

- Effective index structure is crucial

- Documents containing a specific term are located using an **inverted index**
  - Each keyword maps to a list of documents that contain it.

- How to support **or/and** semantics?
  - **OR**: compute union of sets
  - **AND**: compute intersection of sets

Small Example

- Document #1:

  The University of Pittsburgh is located in Pittsburgh.

- Document #2:

  Carnegie Mellon University is located in Pittsburgh.

- Document #3

  Pittsburgh was voted most livable city. Steelers. Steelers!

- Document #4:

  The Steelers won over the Cleveland Browns.

- Document #5:

  The Pittsburgh Steelers have won 6 Super Bowls.

- Document #6:

  Cleveland is located in Ohio.

Preprocessing – stop-word removal

- Document #1:

  ~~The~~ University ~~of~~ Pittsburgh ~~is~~ located ~~in~~ Pittsburgh.

- Document #2:

  Carnegie Mellon University ~~is~~ located ~~in~~ Pittsburgh.

- Document #3

  Pittsburgh ~~was~~ voted ~~most~~ livable city. Steelers. Steelers!

- Document #4:

  ~~The~~ Steelers won ~~over the~~ Cleveland Browns.

- Document #5:

  ~~The~~ Pittsburgh Steelers ~~have~~ won 6 Super Bowls.

- Document #6:

  Cleveland ~~is~~ located ~~in~~ Ohio.

Preprocessing – lower case

- Document #1:

  university pittsburgh located pittsburgh
- Document #2:

  carnegie mellon university located pittsburgh
- Document #3

  pittsburgh voted livable city steelers steelers
- Document #4:

  steelers won cleveland browns.
- Document #5:

  pittsburgh steelers won 6 super bowls
- Document #6:

  cleveland located ohio

Preprocessing – stemming

- Document #1:

  university pittsburgh locat pittsburgh

- Document #2:

  carnegie mellon university locat pittsburgh

- Document #3

  pittsburgh vot livable city steeler steeler

- Document #4:

  steeler won cleveland brown

- Document #5:

  pittsburgh steeler won 6 super bowl

- Document #6:

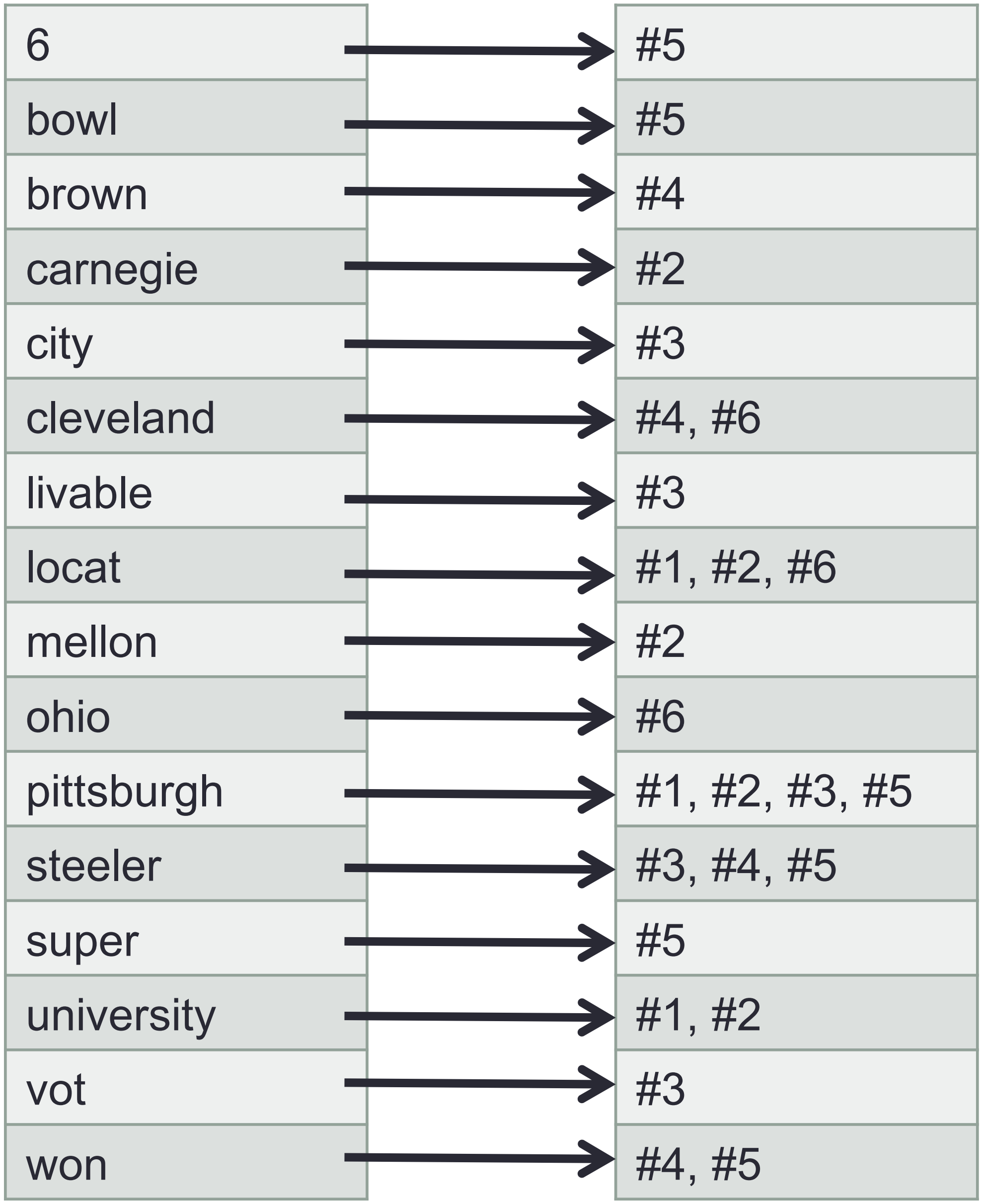
  cleveland locat ohio

Inverted Index Example 1

	#1	#2	#3	#4	#5	#6
6						
bowl						
brown						
carnegie						
city						
cleveland						
livable						
locat						
mellon						
ohio						
pittsburgh						
steeler						
super						
university						
vot						
won						

Inverted Index Example 2

More efficient approach, that considers sparsity.



6	#5
bowl	#5
brown	#4
carnegie	#2
city	#3
cleveland	#4, #6
livable	#3
locat	#1, #2, #6
mellon	#2
ohio	#6
pittsburgh	#1, #2, #3, #5
steeler	#3, #4, #5
super	#5
university	#1, #2
vot	#3
won	#4, #5

Inverted Index Example 3

Store frequency counts for each (keyword, document) combination

	#1	#2	#3	#4	#5	#6
6					1	
bowl					1	
brown				1		
carnegie		1				
city			1			
cleveland				1		1
livable			1			
locat	1	1				1
mellon		1				
ohio						1
pittsburgh	2	1	1		1	
steeler			2	1	1	
super					1	
university	1	1				
vot			1			
won				1	1	

Inverted Index Example 4

Store frequency counts for each (keyword, document) combination

6	(#5, 1)
bowl	(#5, 1)
brown	(#4, 1)
carnegie	(#2, 1)
city	(#3, 1)
cleveland	(#4, 1), (#6, 1)
livable	(#3, 1)
locat	(#1, 1), (#2, 1), (#6, 1)
mellon	(#2, 1)
ohio	(#6, 1)
pittsburgh	(#1, 2), (#2, 1), (#3, 1), (#5, 1)
steeler	(#3, 2), (#4, 1), (#5, 1)
super	(#5, 1)
university	(#1, 1), (#2, 1)
vot	(#3, 1)
won	(#4, 1), (#5, 1)