



Inspire...Educate...Transform.

## TEXT MINING : DAY 2

Dr. Manoj Duse  
Mar 10, 2019



# Agenda

- Performance Measures
- SVD, LSI/LSA,
- Efficient approaches for ranking
- Page Rank and alternative approaches
- Introduction to Semantic Analysis

# Dataset



Example of text data: Titles of Some Technical Memos

- c1: *Human machine interface for 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, ordered 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 et al., 1990; Landauer & Dumais

# Human-User and Human-Minor

	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

$$\underline{r}(\text{human.user}) = -.38$$

$$\underline{r}(\text{human.minors}) = -.29$$

How LSA induces similarity relations by changing estimated entries up or down.

Compare the  $r$  with previous [original data] and  $r$  with this reduced-dim data

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

$$\underline{r}(\text{human.user}) = .94$$

$$\underline{r}(\text{human.minors}) = -.83$$

Reduced Rank  
Approximation

# Shaded and/or boxed rows for the words human , user and minors

- ❑ In this context, minor is a technical term from graph theory.
- ❑ In the original, human never appears in the same passage with either user or minors — they have no co-occurrences.
- ❑ The correlations (using Spearman) are  $-.38$  between human and user, and  $-.29$  between human and minors.
- ❑ In the reconstructed two-dimensional approximation, both have been greatly altered:
  - ❑ human-user correlation has gone up to  $.94$ ,
  - ❑ human-minors correlation down to  $-.83$ .
- ❑ Thus, because the terms human and user occur in contexts of similar meaning—even though never in the same passage—the reduced dim solution represents them as more similar, while the opposite is true of human and minors

# Raw data – document correlations

Correlations between titles in raw data:

	c1	c2	c3	c4	c5	m1	m2	m3
c2	-0.19							
c3	0.00	0.00						
c4	0.00	0.00	0.47					
c5	-0.33	0.58	0.00	-0.31				
m1	-0.17	-0.30	-0.21	-0.16	-0.17			
m2	-0.26	-0.45	-0.32	-0.24	-0.26	0.67		
m3	-0.33	-0.58	-0.41	-0.31	-0.33	0.52	0.77	
m4	-0.33	-0.19	-0.41	-0.31	-0.33	-0.17	0.26	0.56

Correlation among  
HCI related titles  
[c1...c4]

0.02  
-0.30 0.44

Correlation among graph titles  
and HCI titles [latent concepts]

Correlation among graph related  
titles [m1...m4]

# In reduced two dimensions...

Correlations in two dimensional space:

c2	0.91							
c3	1.00	0.91						
c4	1.00	0.88	1.00					
c5	0.85	0.99	0.85	0.81				
m1	-0.85	-0.56	-0.85	-0.88	-0.45			
m2	-0.85	-0.56	-0.85	-0.88	-0.44	1.00		
m3	-0.85	-0.56	-0.85	-0.88	-0.44	1.00	1.00	
m4	-0.81	-0.50	-0.81	-0.84	-0.37	1.00	1.00	1.00

Avg Correlation  
among HCI related  
titles [c1...c4]

0.92

-0.72

1.00

Avg Correlation among graph  
titles and HCI titles [two latent  
concepts] .... More contrast !

Avg Correlation among graph  
related titles [m1...m4]



# Binary Term-Document Incidence Matrix

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

# Term-Document Count Matrices

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

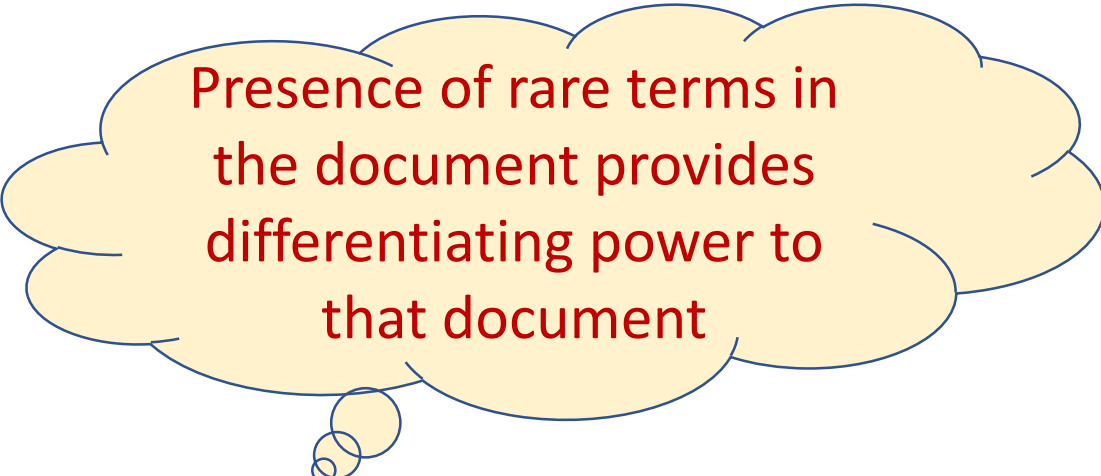
# Tweaking definition of Term Frequency (TF)

- The term frequency  $tf_{t,d}$  of term  $t$  in document  $d$  is defined as the number of times that  $t$  occurs in  $d$ .
- Relevance does not increase proportionally with term frequency.
- So use log frequency weighting
- The log frequency weight of term  $t$  in  $d$  is:  
$$w_{td} = \log_{10} tf_{td} \quad \text{if } tf_{td} > 0;$$

else it is 0.

# Inverse Document Frequency (IDF)

- Frequent terms are less informative than rare terms
- $df_t$  is the document frequency of  $t$ :
  - the number of documents that contain  $t$
  - $df_t \leq N$
- $idf_t = \log_{10} \left( \frac{N}{df_t} \right)$



Presence of rare terms in the document provides differentiating power to that document

# TF-IDF Weighting

$$\text{tf.idf}_{t,d} = \log(1 + \text{tf}_{t,d}) \times \log_{10}(N / \text{df}_t)$$

$$\text{Score}(q, d) = \sum_{t \in q \cap d} \text{tf.idf}_{t,d}$$

- The tf-idf weight of a term is the product of its tf weight and its idf weight.
- Score for a document given a query
- There are variants

# Binary → Count → Weight Matrix

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

Each document is now represented by a real-valued vector of tf-idf weights  $\in \mathbb{R}^{|V|}$

# Quiz

- How does synonymy and polysemy affect similarity scores of documents which contain them ?
- Scent, perfume [used alternatively in documents]
- Fan (fan of actor); fan (ceiling fan)
- Plot (land); plot (of movie); plot (graph)

- Synonymy → Under-estimation
- Polysemy → Over-estimation



Does a very high number of documents in the corpus pose a challenge ?

# Ways to make it more efficient?

So far we have focused on retrieving precisely the  $K$  highest-scoring documents for a query.

What if  $K$  documents that are *likely* to be among the  $K$  highest scoring documents for a query are returned ?

[as long we have not compromised the relevance to a great extent....trade-off]

Objective : dramatically lower the cost of computing

# Elimination Strategy

- Basic algorithm only considers docs containing at least one query term
- Take this further
  - Only consider high-idf [exceeds some thresh-hold] query terms
  - Only consider docs containing many query terms...perhaps all terms?

Inexact Top K

# Champion Lists (aka fancy list or top docs for $t$ )

- Precompute for each term  $t$ , the  $r$  docs of highest weight in  $t$ 's postings
  - This becomes the Champion list for  $t$
- Note that  $r$  has to be chosen at index time
- Value of  $r$  is application dependent
- At query time, only compute scores for docs in the champion list of query term [reduced number of cosine sim calculations and reduced set to choose top  $k$  from]
  - Pick the  $K$  top-scoring docs from amongst these

# Static Quality Scores

- We want top-ranking documents to be both *relevant* and important/*authoritative*
- *Relevance* is being modeled by cosine score : *query-dependent*
  - that is what we have spent time on so far
- *Authority* is typically a query-independent property of a document
  - Wikipedia among websites
  - Certain newspapers
  - A paper with many citations
  - Pagerank – ( we will talk about this soon )
- Assign to each document a *query-independent* quality score in  $[0,1]$   
For each document  $d$  :  $staticQuality(d)$

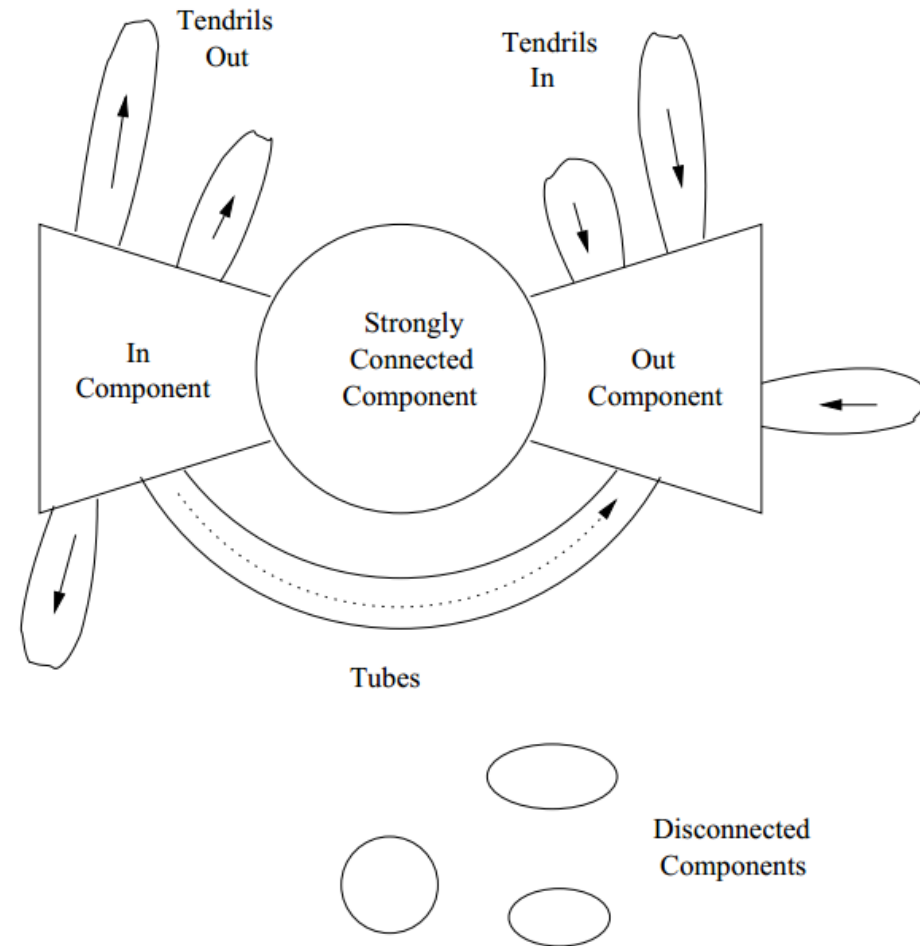
# Net Score

- Consider a simple total score combining cosine relevance and authority
- $\text{net-score}(q,d) = \text{staticQuality}(d) + \text{cosine}(q,d)$ 
  - Can use some other linear combination than an equal weighting
  - Two “signals” of user happiness
- Now we seek the top  $K$  docs by net score [ as before]

# Link Analysis

- How do you think the structure of the web looks like ?

# Bow-tie Structure of the Web





# Spider Traps

- A group of pages is a **spider trap** if there are no links from within the group to outside the group

# Dead End:

- Pages with no out-links are “dead ends” for the random surfer
- Notice the difference between spider traps and dead ends.
- Spider trap pages may link to each other [so there are out-links] but you can not get out of the group

Chakravyuv ?

# Why Page Rank ?

# Why Page Rank

## Relevance vs. Trustworthiness

- User will know whether something is relevant when shown
  - Won't know whether it is trustworthy
- 
- Web content creators are autonomous/uncontrolled
  - Someone may intentionally create pages with keywords just to drive traffic to their page
  - May even use spoofing techniques to show one face to search engine and another to the user
  - World out there is not perfect...

Page rank is best seen as a measure of Importance and Trustworthiness.

# What is a random surfer:

- Surfer who starts at a random page and at any step moves at random to any one of the pages to which the current page links to.
- Spider traps violate the conditions needed for the random walk...
- Random surfer gets trapped and loses the ability to move on to a new random page [randomness is compromised !]
- Limiting [long term steady state] probability of a random surfer being at a given page is Page Rank.

Intuition ?

# Intuition behind Page Rank

- People tend to create links to the pages they think are useful; so random surfers will tend to be at a useful page [eventually] with a higher probability → higher page rank
- A page is important if the probability of a random surfer landing on that page [in long run] is high
- Page Rank  $\Leftrightarrow$  high probability of landing [importance]

Now the mathematics

# Ranking Web Pages

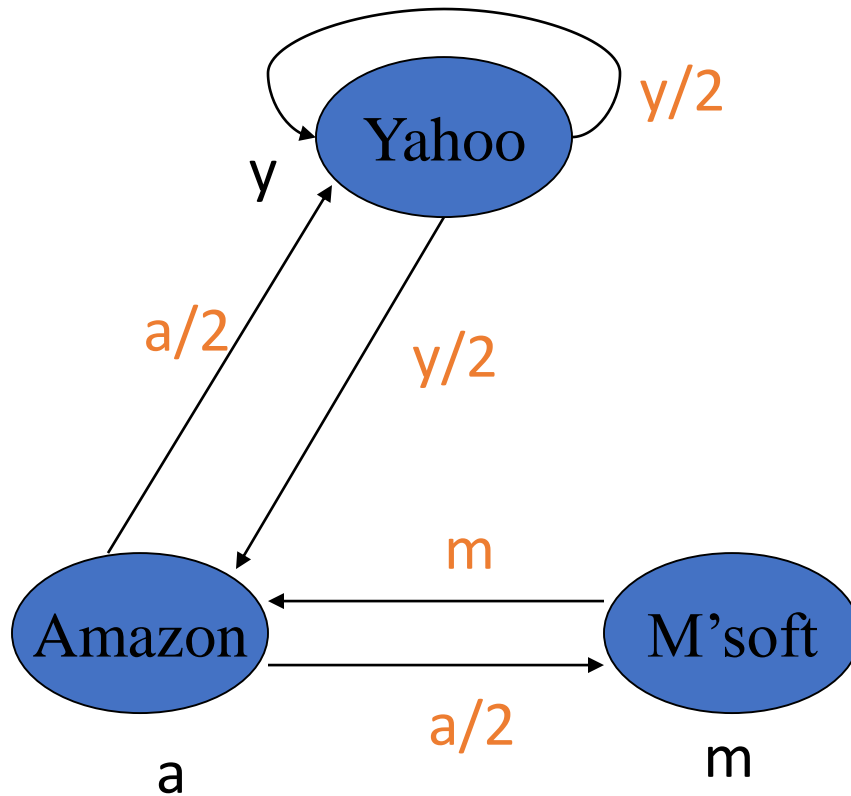
- Web pages are not equally “important”
- In-links as votes
- Are all in-links equal?
  - Each in-link’s vote is proportional to the **importance** of its source page
  - If page **P** with importance **x** has **n** outlinks, each link gets  **$x/n$**  votes
  - Page **P**’s own importance is the sum of the votes on its inlinks

Page is important if important pages link to it....recursive definition!



# Let us work with a small web of 3 pages!

Page Rank views hyper-linked pages as a markov chain



$$y = y/2 + a/2$$

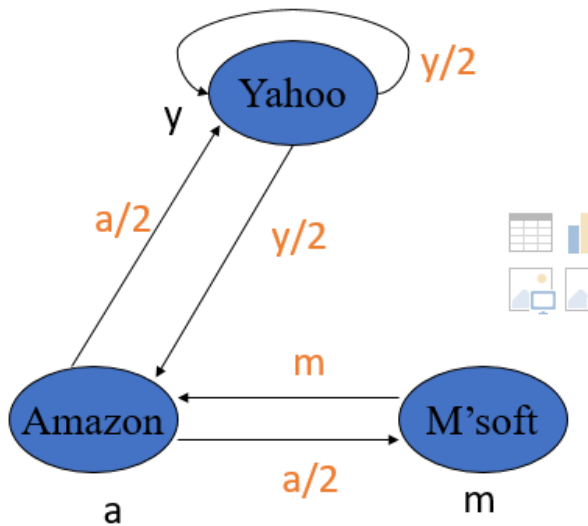
$$a = y/2 + m$$

$$m = a/2$$

# Solving the Equations

- $y+a+m = 1$
  - $y = 2/5,$
  - $a = 2/5,$
  - $m = 1/5$
- 
- Gaussian elimination method works for small examples, but need a better method for large graphs

# Matrix Formulation



$$y = y/2 + a/2$$

$$a = y/2 + m$$

$$m = a/2$$

M=web linkage matrix / transition matrix

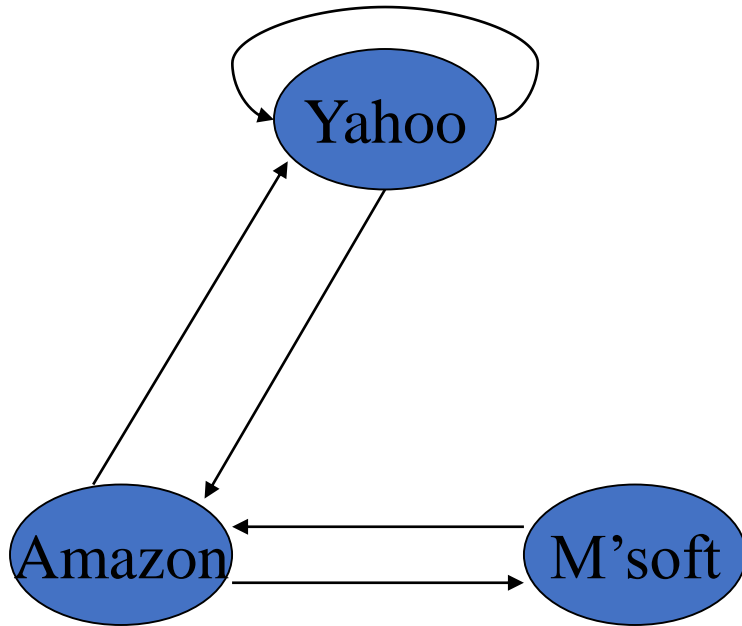
$$r = M * r$$

$$\begin{bmatrix} y \\ a \\ m \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} y \\ a \\ m \end{bmatrix}$$

# Power Iteration Method

- Simple iterative scheme
- Suppose there are  $N$  web pages
- Initialize:  $\mathbf{r}^0 = [1/N, \dots, 1/N]^T$
- Iterate:  $\mathbf{r}^{k+1} = \mathbf{M}\mathbf{r}^k$
- Stop when  $\|\mathbf{r}^{k+1} - \mathbf{r}^k\|_1 < \varepsilon$

# Power Iteration Example



	y	a	m	
y	1/2	1/2	0	1/3
a	1/2	0	1	1/3
m	0	1/2	0	1/3

$$\begin{aligned}
 & [1/2] \times [1/3] + [0] \times [1/3] + [1] \times [1/3] \\
 &= 1/6 + 0 + 1/3 \\
 &= 1/6 + 2/6 = 3/6 = \mathbf{1/2}
 \end{aligned}$$

y		1/3	1/3	5/12	3/8		2/5
a	=	1/3	1/2	1/3	11/24	...	2/5
m		1/3	1/6	1/4	1/6		1/5

# Another iteration

	y	a	m	
y	1/2	1/2	0	1/3
a	1/2	0	1	1/2
m	0	1/2	0	1/6

$$\begin{aligned} & [1/2] \times [1/3] + [0] \times [1/2] + [1] \times [1/6] \\ &= 1/6 + 0 + 1/6 \\ &= 2/6 = 1/3 \end{aligned}$$

→ 5/12  
1/3  
1/4

# Random Walk Interpretation

- At any time  $t$ , surfer is on some page  $P$
  - At time  $t+1$ , the surfer follows an outlink from  $P$  uniformly at random
  - Ends up on some page  $Q$  linked from  $P$
  - Process repeats indefinitely
- 
- Let  $\mathbf{p}(t)$  be a vector whose  $i^{\text{th}}$  component is the probability that the surfer is at page  $i$  at time  $t$ 
    - $\mathbf{p}(t)$  is a probability distribution on pages

# The Stationary Distribution

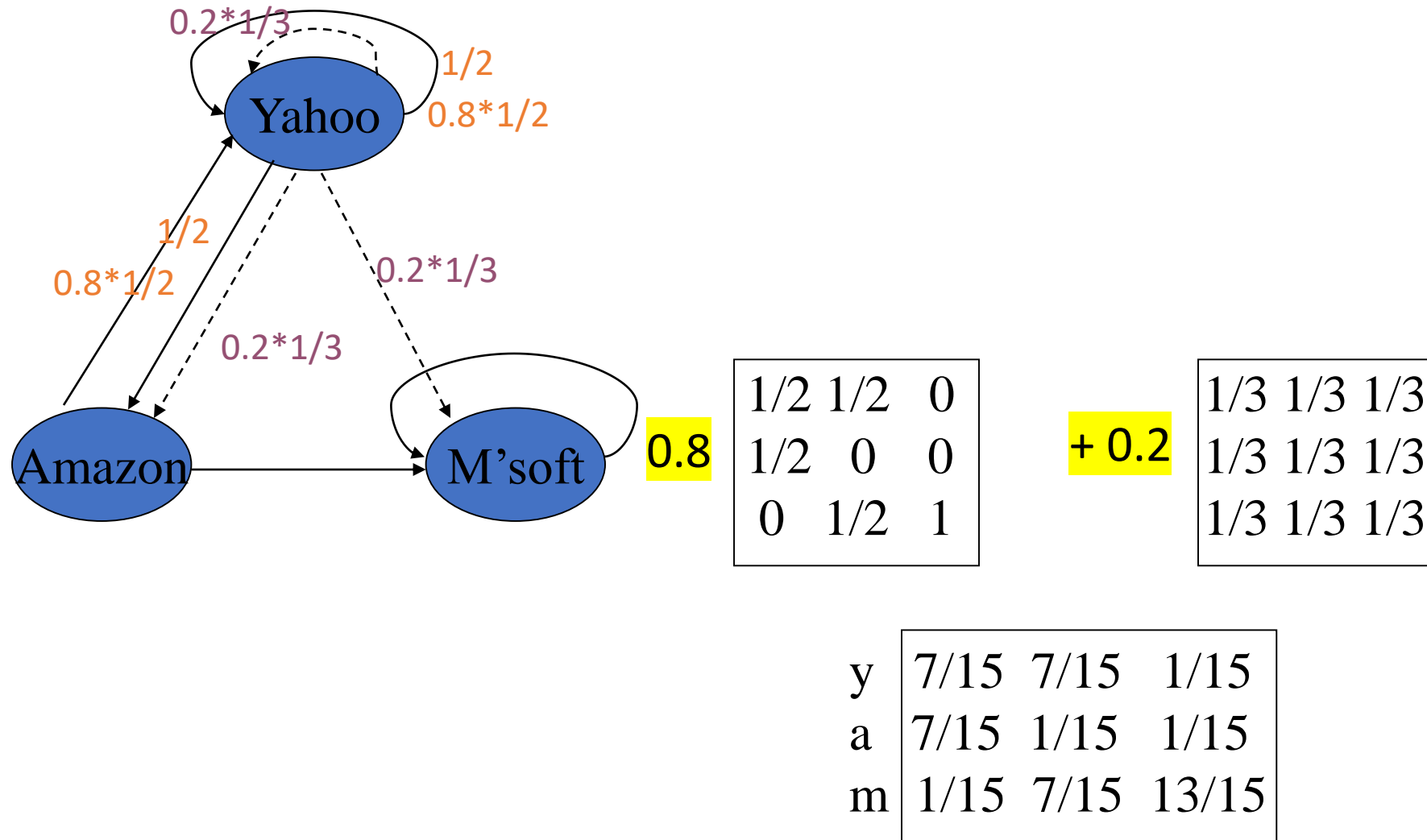
- Where is the surfer at time  $t+1$ ?
  - Follows a link uniformly at random
  - $\mathbf{p}(t+1) = \mathbf{M}\mathbf{p}(t)$
- Suppose the random walk reaches a state such that  $\mathbf{p}(t+1) = \mathbf{M}\mathbf{p}(t) = \mathbf{p}(t)$ 
  - Then  $\mathbf{p}(t)$  is called a **stationary distribution** for the random walk
- Our vector  $\mathbf{r}$  satisfies  $\mathbf{r} = \mathbf{M}\mathbf{r}$ 
  - So it is a stationary distribution for the random surfer



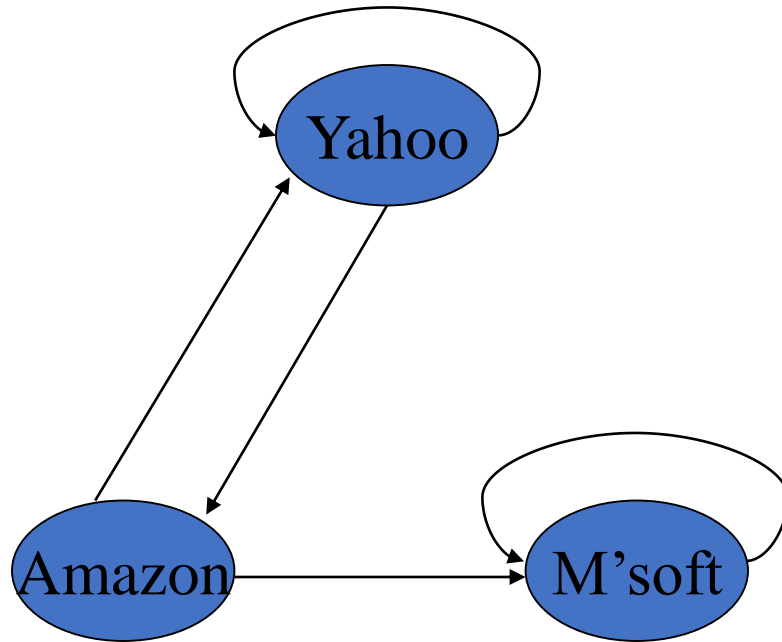
# Dealing with Spider Traps: Random Teleports

- At each time step, the random surfer has two options:
  - With probability  $\beta$ , follow a link at random
  - With probability  $1-\beta$ , jump to some page uniformly at random
  - Common values for  $\beta$  are in the range 0.8 to 0.9
- Surfer will teleport out of spider trap within a few time steps
- Escape path [after struggling for a while !]

# Random Teleports (with $\beta = 0.8$ )



# Random Teleports ( $\beta = 0.8$ )



$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$\begin{matrix} y \\ a \\ m \end{matrix} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

y	1	1.00	0.84	0.776		7/11
a	=	1	0.60	0.60	...	5/11
m		1	1.40	1.56	1.688	21/11

The **PageRank vector**  $\mathbf{r}$  is the stationary distribution of the random walk with teleports

# Topic Sensitive Page Rank

Try to classify users according to the degree of their interest in each of the selected topics:

One useful topic set is the 16 top-level categories (sports, medicine, etc.) of the Open Directory (DMOZ)

Bias the PageRank to favor pages of that topic.

Can create 16 teleport vectors, one for each topic to model the bias.

Figure out what user is interested in, perhaps by the content of the pages they have recently viewed, identify topic from 16, use teleport set for that topic and decide ranking of pages in the same way as earlier.

# Dmoz.org is now closed...mirror available



[Computers](#) > [Artificial Intelligence](#) > [Machine Learning](#) > [Conferences](#)

## ▼ Related categories 2

- [Computers](#) > [Artificial Intelligence](#) > [Conferences and Events](#)
- [Computers](#) > [Artificial Intelligence](#) > [Neural Networks](#) > [Conferences](#)

## ▼ Sites 16

### [Benelearn 2001](#)

Eleventh Dutch-Belgian Conference on Machine Learning. University of Antwerp, Belgium; 21 December 2001.

### [European Conference on Mobile Robots 2005](#)

Ancona, Italy; September 7-10, 2005

### [Fourth International Conference on Intelligent Data Engineering and Automated Learning \(IDEAL2003\)](#)

Hong Kong; March 21-23, 2003. A biennial conference.

$$\text{beta} = 0.8; (1-\text{beta}) = 0.2 \text{ (1/5)}$$

- Teleport set (biased instead of random) let us assume has two pages for the category
- Prob for each page in teleport set =  $\frac{1}{2}$
- [since two pages for the category]
- $\frac{1}{2}$  instead of 3 as it would have been in case of random teleport
- $\frac{1}{10} = \frac{1}{5}$  (this is our 1-beta) \* ( $\frac{1}{2}$  for pages of interest, 0 otherwise]

# Trust Rank

- Improved version of Topic Sensitive Page Rank
- What is improved ?
- Teleport set for a specific topic is now a carefully chosen set of “high trust pages” in that category/topic.
- That makes it ➡ Trust Rank

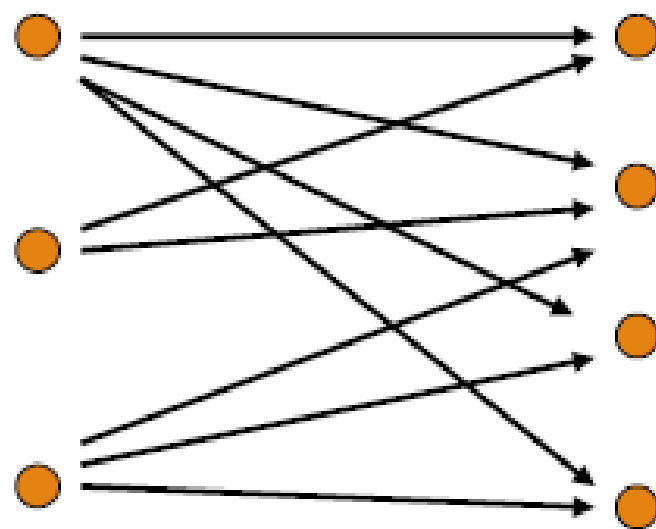
Is there any thing other than Page Rank?

HITS



# Hubs and Authorities

- Also called HITS → Hyperlink Induced Topic Search
- Page Rank : One dimensional importance of web pages
- HITS : Two dimensions of importance
  - Pages that provide valuable content for the topic : Authorities
  - Pages that themselves are not valuable wrt content but they tell you where to go to find right/useful/authentic information : Hubs



hubs

authorities

- Good Hub:

A page is a good hub if it links to good authorities  
...and what is a good authority ?

- Good Authority:

A page is a good authority if it is linked to by good hubs

Recursive Definitions

- Hubbiness proportional to sum of its successors [authorities]
- Authority proportional to sum of its predecessors [hubs]
- HITS convergence not susceptible to dead ends and spider traps

# Sentiment Analysis

# Sentiment Analysis

Many Names:

- Opinion Mining
- Sentiment Mining
- Subjectivity analysis
- Sentiment Classification

# What is Sentiment

- Sentiment  $\Leftrightarrow$  feelings
  - Attitudes
  - Emotions
  - Opinions
- May be function of time [state of mind] !

# What is Sentiment Analysis

- Classify given text based on the overall sentiments expressed by the author
- Use NLP, statistics, or machine learning methods to extract, identify, or characterize the sentiment content of a text unit
- Different levels
  - Document
  - Sentence
  - Feature
- Classification levels
  - Binary
  - Multi Class



# Basic Components

- Opinion Holder – Who is talking ?
- Object – Item on which opinion is expressed.
- Opinion – Attitude or view of the opinion holder.

# Motivation [use cases]

## **Product review mining:**

What features of a new product do customers like and which features do they dislike?

## **Review classification:**

Is a review [of a movie, or a book] positive or negative ?

## **Tracking sentiments toward topics over time:**

Are people supporting Dhoni for inclusion in next world cup?

How has sentiments on Demonetization changed over time?

# More use cases

- Is the customer email reflects satisfaction or otherwise?
- How is the response to ad campaign based on tweeter activity?
- Why aren't consumers buying our products...more so in B2C than B2B
- Valuable feedback/opinions of those who did not buy [not in the set of current customers]

# Document Level Analysis

- Documents can be reviews, blog posts, ..
- Assumption:
  - Each document focuses on a single object/topic.
  - Only single opinion holder.
- Task : Determine the overall sentiment of the writer expressed in the document.

# Sentence Level Analysis

- Considers each sentence as a separate unit.
- *Assumption* : sentence contain only one opinion.
- *Task 1*: identify if sentence is subjective or objective
- *Task 2*: identify polarity of sentence.

# Feature Level

- Task 1: identify and extract object features
  - Task 2: determine polarity of opinions on features
  - Task 3: group same features
  - Task 4: summarization
- 
- Ex. This mobile has *good* camera but *poor* battery life.

# Sentiment Lexicons

- Positive words: abide, ability, able, absolve, appreciate, enjoy
- Negative words: abandon, abdicate, deteriorate, hate, tedious
- Available lexicons
  - [General Inquirer](#): 1915 positive words and 2291 negative words
  - [LIWC \(Linguistic Inquiry and Word Count\)](#): 2300 words, >70 classes
  - [MPQA Subjectivity Cues Lexicon](#): 2718 positive words and 4912 negative words. Each word annotated for intensity (strong, weak)
  - [Bing Liu Opinion Lexicon](#): 2006 positive words and 4783 negative words
  - [SentiWordNet](#)

# SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining

- SentiWordNet assigns to each synset of WordNet three sentiment scores:
  - positivity,
  - negativity,
  - objectivity.
- The current "official" version of SentiWordNet is 3.0, which is based on WordNet 3.0.



# Approaches: Evaluating sentence polarity

Extract “opinion sentences” based on the presence of a predetermined list of product features and adjectives

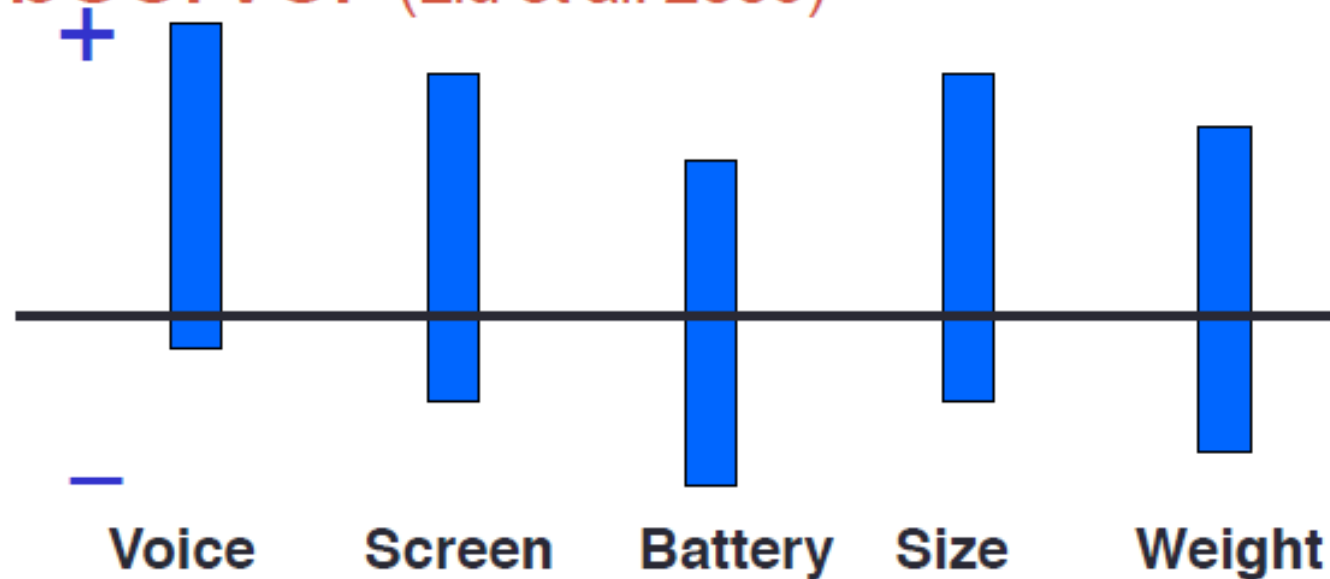
“The lens is excellent”

Evaluate the sentences based on counts of positive vs negative polarity words

Summarize

# Opinion Observer (Liu et al. 2005)

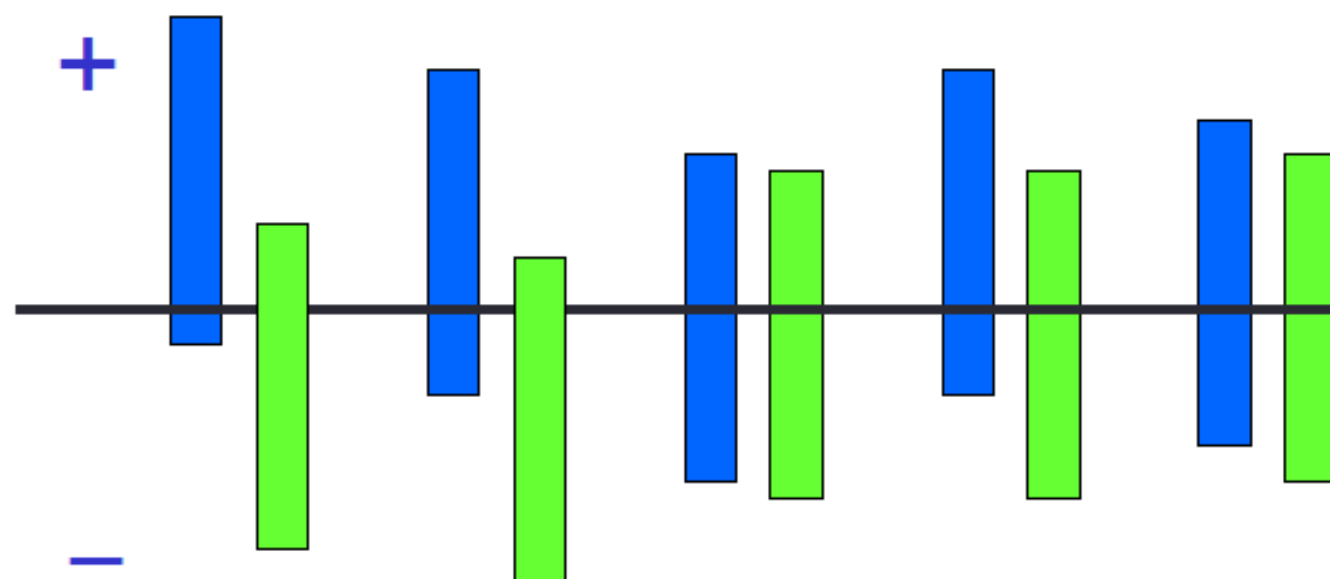
Summary of  
reviews of  
Cell Phone 1



Comparison of  
reviews of

Cell Phone 1

Cell Phone 2



# Sentiment classification is essentially a text classification problem.

Traditional text classification mainly classifies documents of different topics, e.g., politics, sciences, and sports. In such classifications, topic related words are the key features.

However, in sentiment classification, sentiment or opinion words that indicate positive or negative opinions are more important, e.g., *great*, *excellent*, *amazing*, *horrible*, *bad*, *worst*, etc.

Since it is a text classification problem, any existing supervised learning method can be applied, e.g., naïve Bayes classification, and support vector machines (SVM). Training datasets of reviews etc are available.

# CHALLENGES

- Ambiguous words
  - This music cd is literal waste of time. (negative)
  - Please throw your waste material here. (neutral)
- Sarcasm detection and handling
  - “That’s great! All the features I wanted - too bad they don’t work. ”
  - What a great car! It didn’t start on the first day .
- Handling negation
  - This is not a good book to read for beginners

# Recap



Web: <http://www.insofe.edu.in>

Facebook: <https://www.facebook.com/insofe>

Twitter: <https://twitter.com/Insofeedu>

YouTube: <http://www.youtube.com/InsofeVideos>

SlideShare: <http://www.slideshare.net/INSOFE>

LinkedIn: <http://www.linkedin.com/company/international-school-of-engineering>