CAT3024N Intelligent Systems

Information Retrieval

Why Search?

- A billion or so searches per day…
- Boost to productivity
  - Intellectual & economic
- Search is (*still*) ‘hot’
  - Google, Amazon, Ebay,
  - Search for/in books, products, music, people, …
- Fascinating research problem.
- You can learn to be a something of a search expert in one quarter!

What is “Information Extraction”

As a task: **Filling slots in a database from sub-segments of text.**

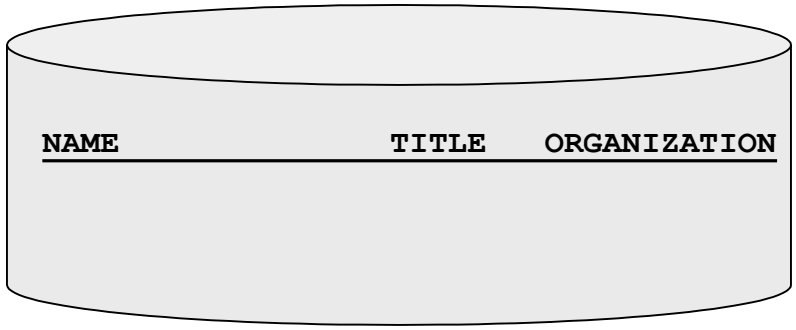
October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access.“

Richard Stallman, founder of the Free Software Foundation, countered saying…



Slides from Cohen & McCallum

What is “Information Extraction”

As a task: **Filling slots in a database from sub-segments of text.**

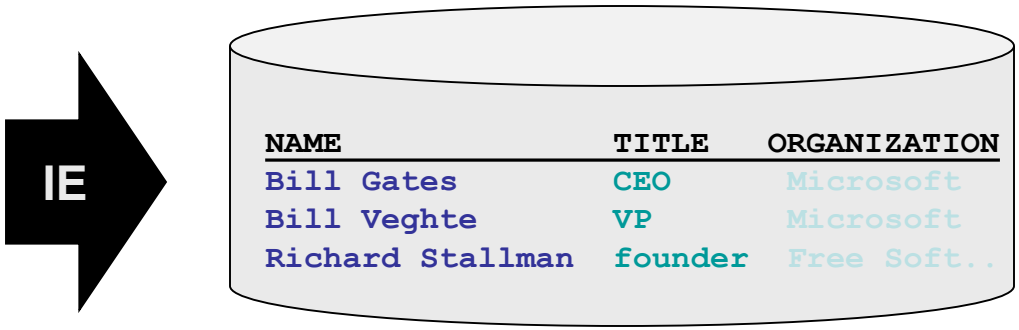
October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access.“

Richard Stallman, founder of the Free Software Foundation, countered saying…

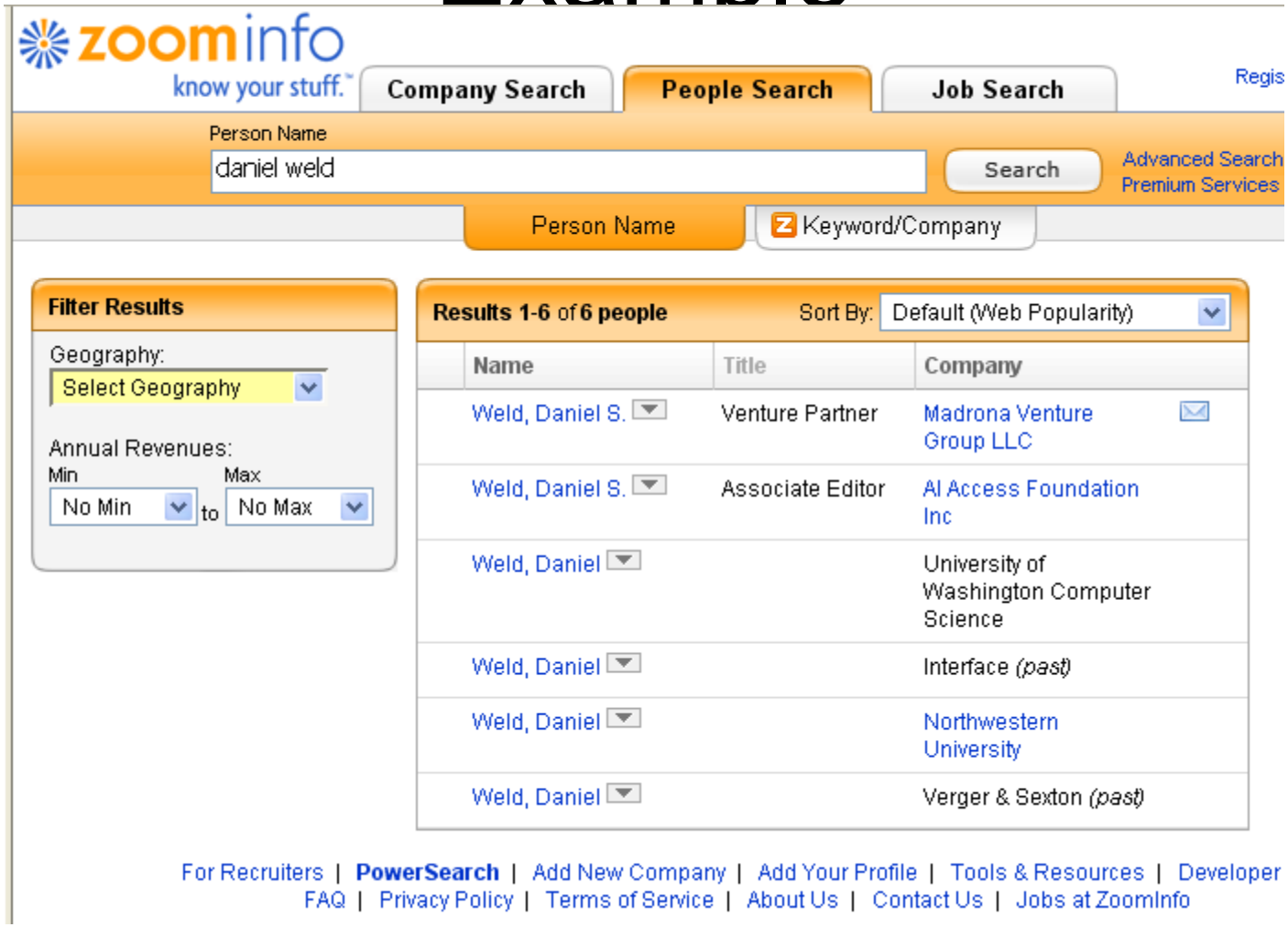


Slides from Cohen & McCallum

Why Information Extraction

- Next-Generation Search
  - People
    - Zoominfo
    - Flipdog
    - Intellius
  - Research Papers
    - Citeseer
    - Google scholar
  - Product search
- Question Answering

Example


zoominfo
know your stuff.
Company Search
People Search
Job Search
Regis
Person Name
daniel weld
Search
Advanced Search
Premium Services
Person Name
Keyword/Company
Filter Results
Geography:
Select Geography
Annual Revenues:
Min
Max
No Min
to
No Max
Results 1-6 of 6 people
Sort By:
Default (Web Popularity)
Name
Title
Company
Weld, Daniel S.
Venture Partner
Madrona Venture Group LLC
Weld, Daniel S.
Associate Editor
AI Access Foundation Inc
Weld, Daniel
University of Washington Computer Science
Weld, Daniel
Interface (past)
Weld, Daniel
Northwestern University
Weld, Daniel
Verger & Sexton (past)
For Recruiters | PowerSearch | Add New Company | Add Your Profile | Tools & Resources | Developer
FAQ | Privacy Policy | Terms of Service | About Us | Contact Us | Jobs at ZoomInfo

…Continued

Dr. Daniel S. Weld This is Me

Venture Partner

Madrona Venture Group LLC

Contact this person

This profile was automatically generated using **47 references** found on the Internet. This information has not been verified. Learn more...

Employment History

Venture Partner[2]
Madrona Venture Group LLC

Headquarters Address:
1000 Second Avenue Suite 3700
Seattle, WA 98104
USA

Website: www.madrona.com
Phone: (206) 674-3000
Fax: (206) 674-8703

Madrona Venture Group is a leading venture capital firm in the Pacific Northwest with an investment strategy focused on early-stage technology companies.

WRF and TJ Cable Professor of Computer Science and Engineering[2]
University of Washington

11/21/2020 10:07 AM

Web References

View all 47 references

1. www.madronagroup.com
www.madronagroup.com/team/vpar
- [Cached]
Published on: 3/14/2007 Last Visited: 3/14/2007

Daniel S. Weld
...
Daniel S. Weld Email Dan Daniel S. Weld is a Venture Partner with Madrona and the WRF/TJ Cable Professor of Computer Science & Engineering at the University of Washington. Dr. Weld co-founded Netbot Inc. (acquired by Excite), AdRelevance (acquired by Nielson Netratings) and Nimble Technology (acquired by Actuate). In addition, he serves as a member of Madrona's Technology Advisory Board.

Dr. Weld received a Presidential Young Investigator's award in

Member, Computer Science and Engineering Department[3]
University of Washington

Member of the Faculty of Computer Science and Engineering[4]
University of Washington

Chief Scientist[5]
AdRelevance Inc

Founder[2]
Netbot Inc

Co-Founder[2]
Journal of AI Research

Program Chair
American Association for Artificial Intelligence

Board Membership and Affiliations

Member of Technology Advisory Board[2]
Madrona Venture Group LLC

Architecture Committee Member
DAML

Fellow (past)[4]
American Association for Artificial Intelligence

1989, an Office of Naval Research Young Investigator's award in 1990, was elected a AAAI Fellow in 1999, and an ACM Fellow in 2006. He earned bachelor's degrees in both Computer Science and Biochemistry at Yale University in 1982, and a Ph.D. from the Massachusetts Institute of Technology Artificial Intelligence Lab in 1988. Dr. Weld is co-founder of the Journal of AI Research and is on the Editorial Board of Artificial Intelligence.

2. www.madrona.com
www.madrona.com/team/vpandof.h
- [Cached]
Published on: 3/14/2007 Last Visited: 3/14/2007

Daniel S. Weld
...
Daniel S. Weld Email Dan Daniel S. Weld is a Venture Partner with Madrona and the WRF/TJ Cable Professor of Computer Science & Engineering at the University of Washington. Dr. Weld co-founded Netbot Inc. (acquired by Excite), AdRelevance (acquired by Nielson Netratings) and Nimble Technology (acquired by Actuate). In addition, he serves as a member of Madrona's Technology Advisory Board.

Dr. Weld received a Presidential Young Investigator's award in 1989, an Office of Naval Research Young Investigator's

…Continued Some More

Education

bachelor's degrees, Computer Science and Biochemistry[1]
Yale University

Ph.D.[1]
Massachusetts Institute of Technology Artificial Intelligence Lab

4. Nimble Technology: Tech. Advisory Board - provides XML data integration software for real-time unified views of database, data warehouse, and unstructured sources. Create enterprise information portals, business intelligence and other applications.
www.nimble.com/company/advisor - [Cached]
Published on: 4/7/2004 Last Visited: 4/7/2004

5. AdRelevance - press releases
intelligence.adrelevance.com/p - [Cached]
Published on: 11/4/2002 Last Visited: 7/19/2003

The OMNIAC technology was developed by a team of engineers led by AdRelevance Chief Scientist, Dan Weld, Ph.D. and Vice President of Engineering Jay Bartot.
...
Weld and Bartot are probably best known for their work in bringing Jango, NetBot's intelligent shopping agent, to market in 1997.

CiteSeer *vs.* Scholar

CiteSeer Scientific Literature Digital Library Find: Daniel Weld [Documents] [Citations]

Searching for **PHRASE daniel weld**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Googl
MSN CSB DBLP
35 documents found. **Order: number of citations.**

A Softbot-Based Interface to the Internet - Etzioni, Weld (1994) (Correct) (151 citations)
Interface to the Internet Oren Etzioni and **Daniel Weld** Department of Computer Science and
the AAAI spring symposium on software agents. **Daniel Weld** received bachelor's degrees in both Computer
mobile.csie.ntu.edu.tw/~yjhsu/courses/u1760/papers/Etzioni/softbot-cacm.ps.gz

An Algorithm for Probabilistic Planning - Kushmerick, Hanks, Weld (1993) (Correct) (134 citations)
Planning 3 Nicholas Kushmerick Steve Hanks **Daniel Weld** Department of Computer Science and
[14] Nicholas Kushmerick, Steve Hanks, and **Daniel Weld**. An algorithm for probabilistic planning.
www.cs.washington.edu/homes/weld/papers/tr93-06-03.pdf

Probabilistic Planning with Information Gathering and.. - Draper, Hanks, Weld (1994) (Correct) (122 citations)
Execution 3 Denise Draper Steve Hanks **Daniel Weld** Department of Computer Science and
Contingent Execution Denise Draper, Steve Hanks, Daniel Wel
ftp.cs.washington.edu/pub/ai/cbur-aips94.ps.Z

An Approach to Planning with Incomplete Information - Etzioni, Hanks, Weld.. (1992) (Correct) (101 citations)
Appears in KR-92 Oren Etzioni, Steve Hanks, **Daniel Weld**, Denise Draper, Neal Lesh, Mike Williamson 3
1971. Hanks and Weld 1992] Steven Hanks and **Daniel Weld**. Systematic adaptation for case-based
ftp.cs.washington.edu/pub/ai/etzioni/softbots/kr92.ps.Z

An Approach to Planning with Incomplete.. - Etzioni, Hanks.. (1992) (Correct) (101 citations)
(Extended Abstract) Oren Etzioni, Steve Hanks, **Daniel Weld**, Denise Draper, Neal Lesh, Mike Williamson 3
1971. Hanks and Weld 1992] Steven Hanks and **Daniel Weld**. Systematic adaptation for case-based
ftp.cs.washington.edu/pub/ai/uwl-kr92.ps.Z

An Adaptive Query Execution System for Data Integration - Ives, Florescu, Friedman, .. (1999) (Correct) (71 citat
Daniela Florescu, Marc Friedman, Alon Levy, Daniel Wel
data.cs.washington.edu/papers/ifflw_sigmod99.ps

An Algorithm for Probabilistic Least-Commitment Planning - Kushmerick, Hanks, Weld (1994) (Correct) (56 citati
Planning 3 Nicholas Kushmerick Steve Hanks **Daniel Weld** Department of Computer Science and
Planning Nicholas Kushmerick, Steve Hanks, Daniel Wel
www.cs.washington.edu/homes/weld/papers/buridan-aaai94.pdf

INTELLIGENT AGENTS ON THE INTERNET: Fact, Fiction, and Forecast - Etzioni, Weld (1995) (Correct) (55 cit
Fact, Fiction, and Forecast Oren Etzioni &**Daniel** S. **Weld** 3 Department of Computer Science and
Fiction, and Forecast Oren Etzioni &**Daniel** S. **Weld** 3 Department of Computer Science and
of Washington Seattle, WA 98195-2350 fetzioni, **weld**g@cs.washington.edu May 30, 1995 Abstract Computer
ftp.cs.washington.edu/pub/ai/ieee-expert.ps.Z

Google Scholar BETA Web Images Video News Map
Daniel Weld

Scholar All articles - Recent articles

All Results
D Weld
O Etzioni
R Doorenbos
Z Ives
J Penberthy

A scalable comparison-shopping agen
RB Doorenbos, O Etzioni, DS **Weld** - Procee
cs.washington.edu
The World- Wide- Web is less agent-friendly t
on the Web is presented in loosely structured
agent-readable semantics. HTML annotations
Cited by 468 - Related Articles - View as HTM

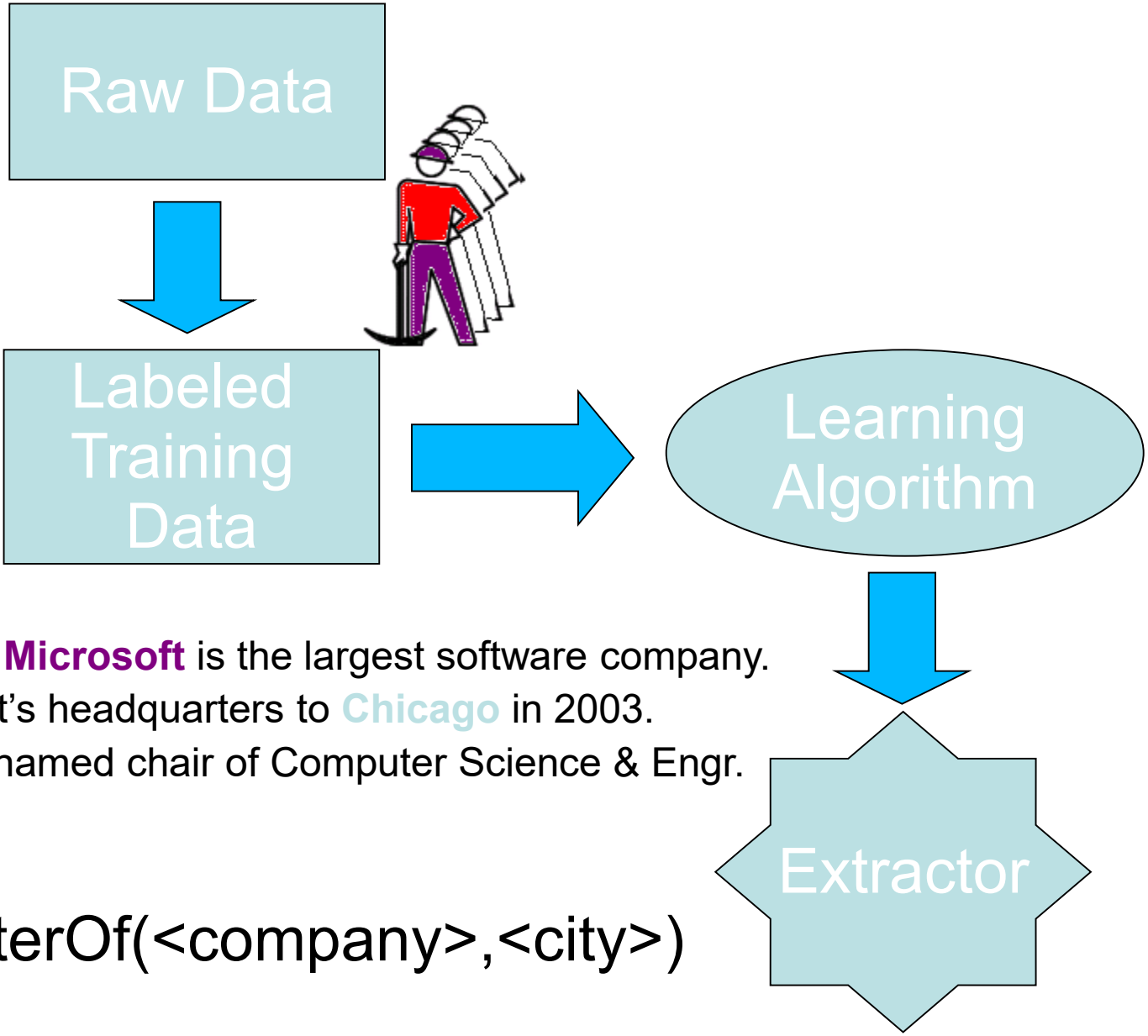
A Softbot-Based Interface to the Intern
O Etzioni, D **Weld** - Communications of the A
Page 1. A Softbot-Based Interface to the Inter
Department of Computer Science and Engine
WA 98195 f etzioni, **weld** g @cs.washington.
Cited by 452 - Related Articles - View as HTM

UCPOP: A sound, complete, partial or
JS Penberthy, D **Weld** - Proceedings of the T
We describe the ucpop partial order plan- ning
of Pednault's ADL action representation. In pa
actions that have conditional e ects, universal
Cited by 450 - Related Articles - View as HTM

An Introduction to Least Commitment P
DS **Weld** - AI Magazine, 1994 - cs.nmsu.edu
T o achieve their goals, agents often need to a
be no surprise that the quest of building intelli
researchers to investigate algorithms for gene
Cited by 377 - Related Articles - Web Search

Readings in qualitative reasonin

Traditional, Supervised I.E.



✓ Kirkland-based **Microsoft** is the largest software company.
✓ **Boeing** moved it's headquarters to Chicago in 2003.
✗ Hank Levy was named chair of Computer Science & Engr.

…

HeadquarterOf(<company>,<city>)

Kylin: Self-Supervised Information Extraction from Wikipedia

[Wu & Weld CIKM 2007]

From infoboxes to a training set

Clearfield County, Pennsylvania	
Statistics	
Founded	March 26, 1804
Seat	Clearfield
Area	
- **Total**	2,988 km² (1,154 mi²)
- **Land**	sq mi (km²)
- **Water**	17 km² (6 mi²), 0.56%
Population	
- **(2000)**	83,382
- **Density**	28/km²

Clearfield County was created in [redacted] from parts of Huntingdon and Lycoming Counties but was administered as part of Centre County until 1812.

Its county seat is [redacted].

2,972 km² (1,147 mi²) of it is land and 17 km² (7 mi²) of it (0.56%) is water.

As of 2005, the population density was 28.2/km².

New York City hotels > Mandarin Oriental New York

Opine

Review Summary

Service quality: excellent (3), good (2), best, professional, better, *view all (*

Service attention: attentive (2)

Room beauty: **absolutely beautiful**, beautiful, *view all (2)*

User comments:

The service was excellent and our room was absolutely beautiful . Read m

When compared to Mandarin Oriental New York, **Room beauty** is

- worse at The Premier (33 others)

Quality: best, finest, love, better, *view all (4)*

Staff courtesy: extremely courteous, courteous, *view all (2)*

Beauty: beautiful

Ancient History

- Pre-history: Dewey Decimal system
  - and other bizarre medieval rituals performed by hand

- 1960: Ted Nelson proposes Xanadu
  - Hyperext vision of WWW---why did it fail?
  - Focus on copyright issues (still a thorny problem)
  - Focus on stable, bidirectional links
  - *"Trying to fix HTML is like trying to graft arms and legs onto hamburger"* -- Ted Nelson

1961 Kleinrock paper on packet switching

Contrast with phone lines - circuit switched.

Paleolithic Era

1965 Gordon Moore proposes law

1966 Design of ARPAnet

1968 Doug Engelbart: the first WIMP

1969 First ARPAnet message UCLA -> SRI

1970 ARPAnet spans country, has 5 nodes

1971 ARPAnet has 15 nodes

1972 First email programs, FTP spec

The Personal Computer Era

1974 Intel launches 8080;

TCP design

1975 Gates/Allen write Basic - Altair 8800

1976 Jobs/Wozniak form Apple Computer

111 hosts on ARPAnet

1979 Visicalc

1981 Microsoft has 40 employees;

IBM PC

1984 Launch of Macintosh

1986 Microsoft goes public

Internet ramps up

1983 ARPAnet uses TCP/IP, Design of DNS
1000 hosts on ARPAnet

1985 Symbolic.com first registered domain name

1989 100,000 hosts on Internet

1990 Cisco Systems goes public
Tim Berners-Lee creates WWW at CERN

Web Search Pre-History

- 1950s: “Information Retrieval” (IR) term coined
- 1960s-70s: SMART system, vector space model,
  - Gerald Salton (Cornell) father of IR
- 1980s: Proprietary document DBs
  - (Lexis-Nexis, Medline)
- 1990: Archie (index file names, anon. ftp)
- 1991: Gopher (menus, links to servers)
- 1992: Veronica (index of menu items on gophers)
- 1993: Jughead (keyword + boolean search)
- Rapid evolution, but what is missing?

Modern History of Search

- 1993: WWW Wanderer (first crawler)
- 1994: WebCrawler, Lycos (1st widely-used SEs)
  - WebCrawler was a UW class project by Brian Pinkerton
- 1994: Yahoo directory (Stanford; founded ’95)

  Amazon founded

  Netscape founded (90% mkt share → 1%
- 1995: Ebay

  MetaCrawler (1st major meta-SE)
  - UW Master’s thesis by Erik Selberg

Discovery of the Biz Model

- 1996: Flash by Macromedia
  - later acquired by Adobe
- 1997: goto.com
  - “sponsored links” pay-per-click
- 1997: AskJeeves (question answering)
- 1997: Netbot
  - comparison-shopping search
- 1998: Open directory launched

  Google, pagerank algorithm

  Paypal founded

Turn of the Millennium

- 1999: IE becomes dominant browser
  Napster starts operation
  Search Engines → portals (Yahoo, Excite)
  "Search is a commodity"
- 2000: Flipdog
  - commercial information extraction)
- 2001: Bittorrent protocol (now 35% of internet)
  Ascendance of Google
  "Search is nirvana"
- 2002: IE peaks at 90% market share

Approaching the Present

- 2003: Skype released
- 2004: Facebook founded

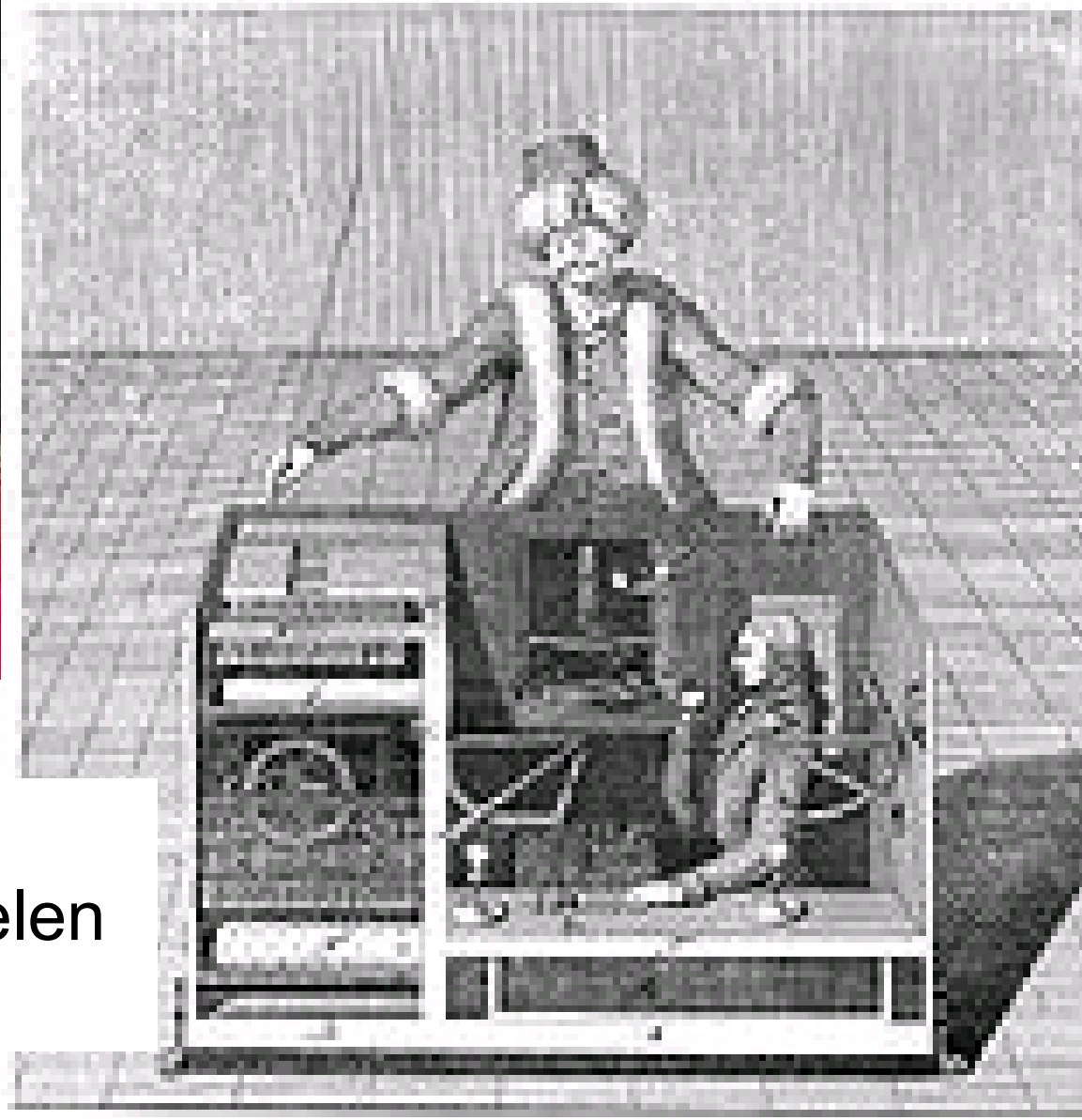
  Social news (Digg)
- 2005: Youtube founded
  - 9.5 B videos shown per month
  - 33 months after founding!
- 2006: Twitter founded
- 2007: Google Streetview

  Apple iPhone
- 2009: Facebook 200M users

Future of the Net

- Domination of Mobile Devices (cellphone, etc)
- Link-Spamming (Arms race to bias SE ranking)
- Local Search, Digital Earth
- Image & Video search
- Social news (Digg / Twitter)
- Crowd Sourcing
- What else?

Mechanical Turk



Built in 1770 by
Wolfgang von Kempelen

- Launched in Nov ’05
  - Initially: detect duplicate product pages
- 100k workers in 100 countries by 3/07
  - 34k HITs on 3/28/08
- Search for Jim Gray
  - 12k searchers

Observations

- Internet/Web *evolved* - it wasn't created

- Scalability beats structure
  - **search engines over directories**
  - **Web over hypertext**

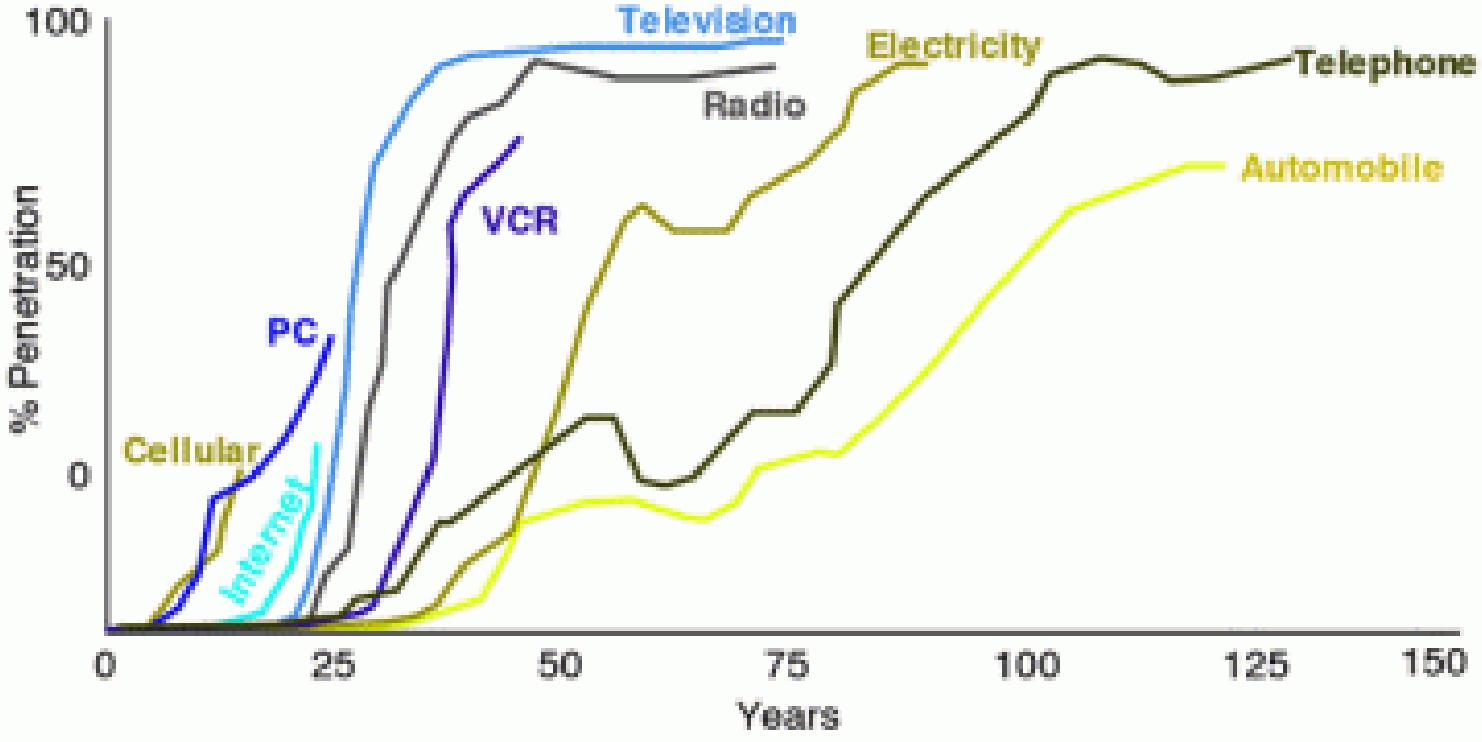
- "We are 10 seconds from the Big Bang"
  - **John Doerr**

Adoption

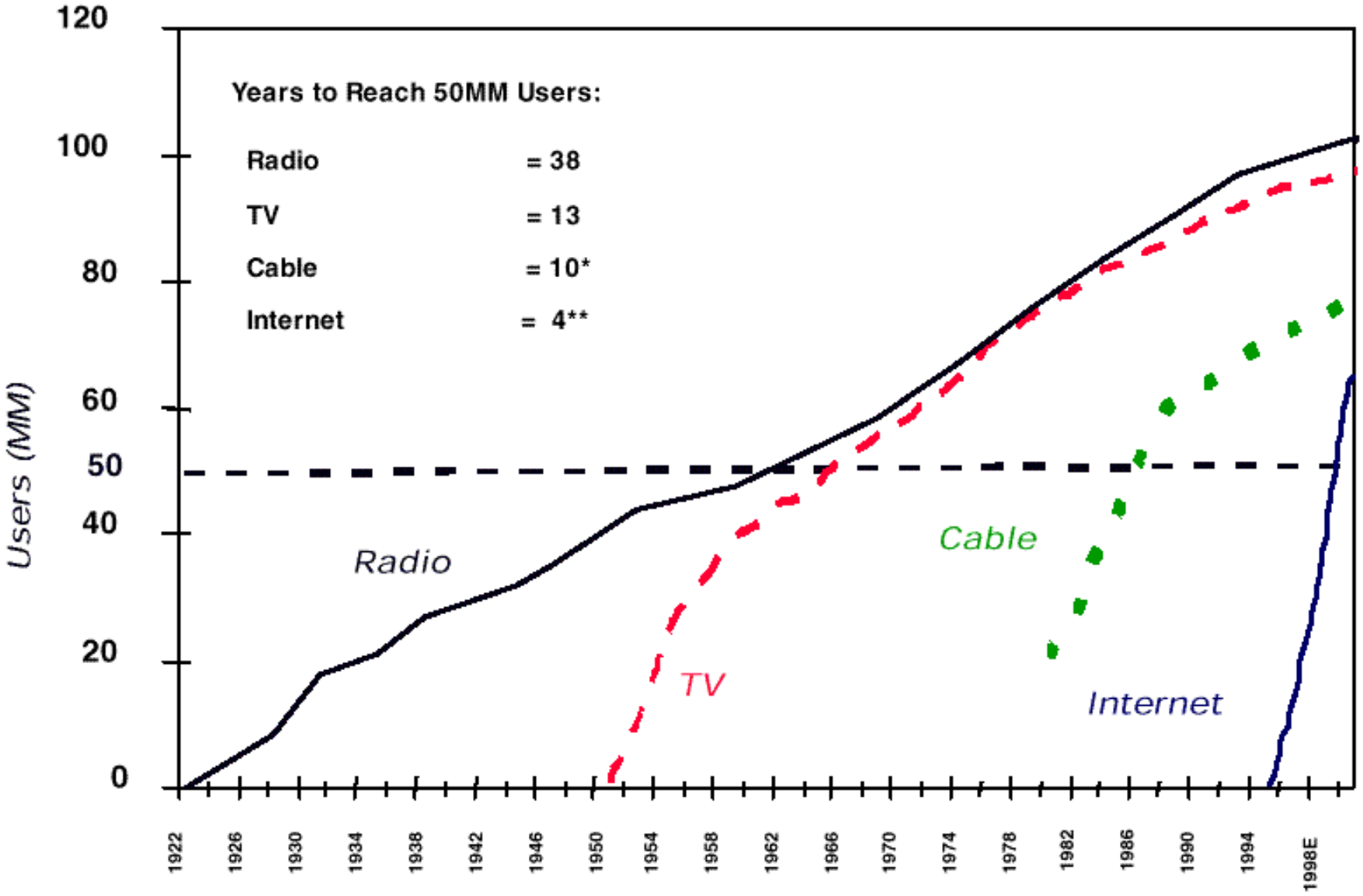
Facilitating Innovation

the pace of innovation is increasing

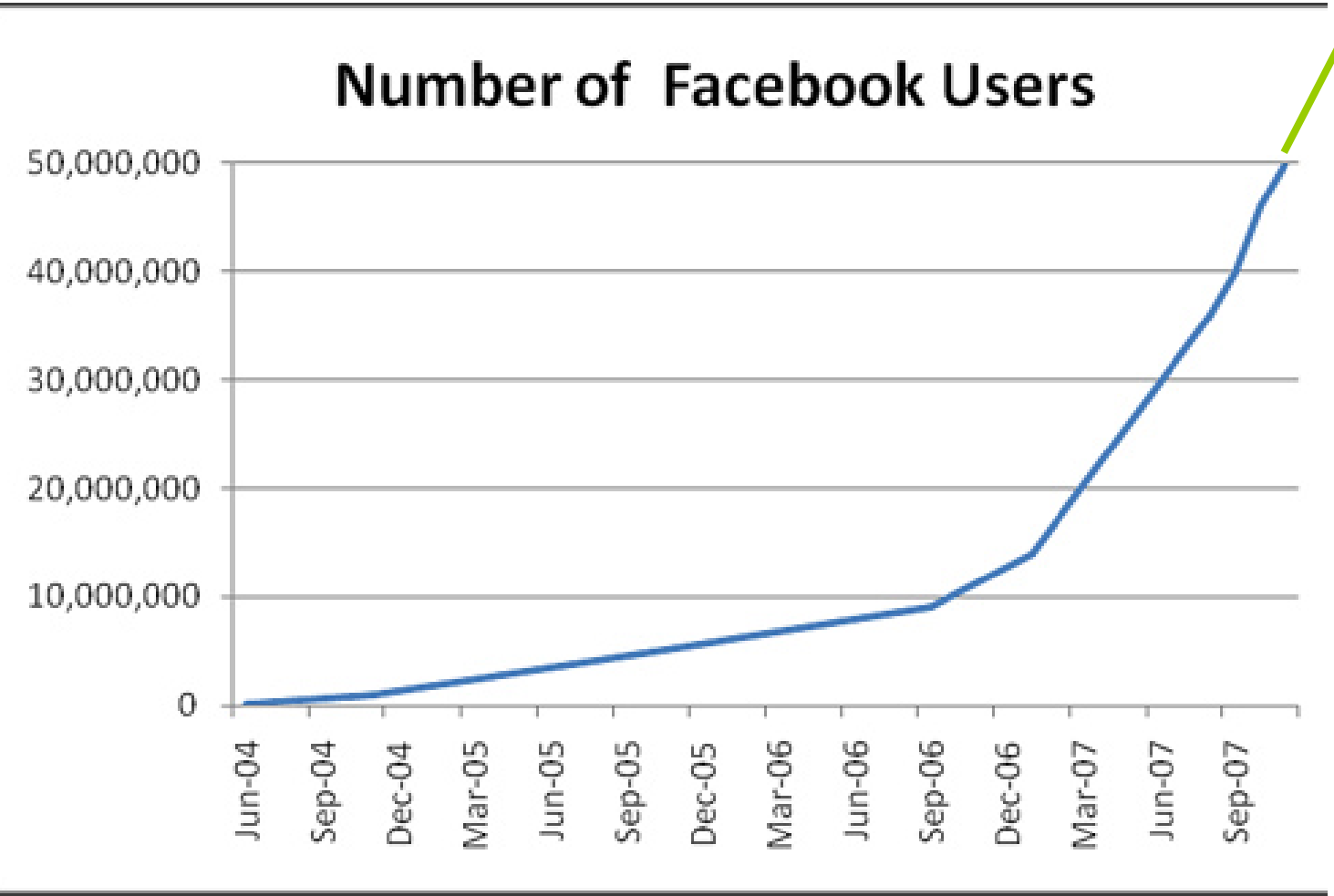
- Newer technologies taking hold at double or triple previous rates



Accelerating



And now?

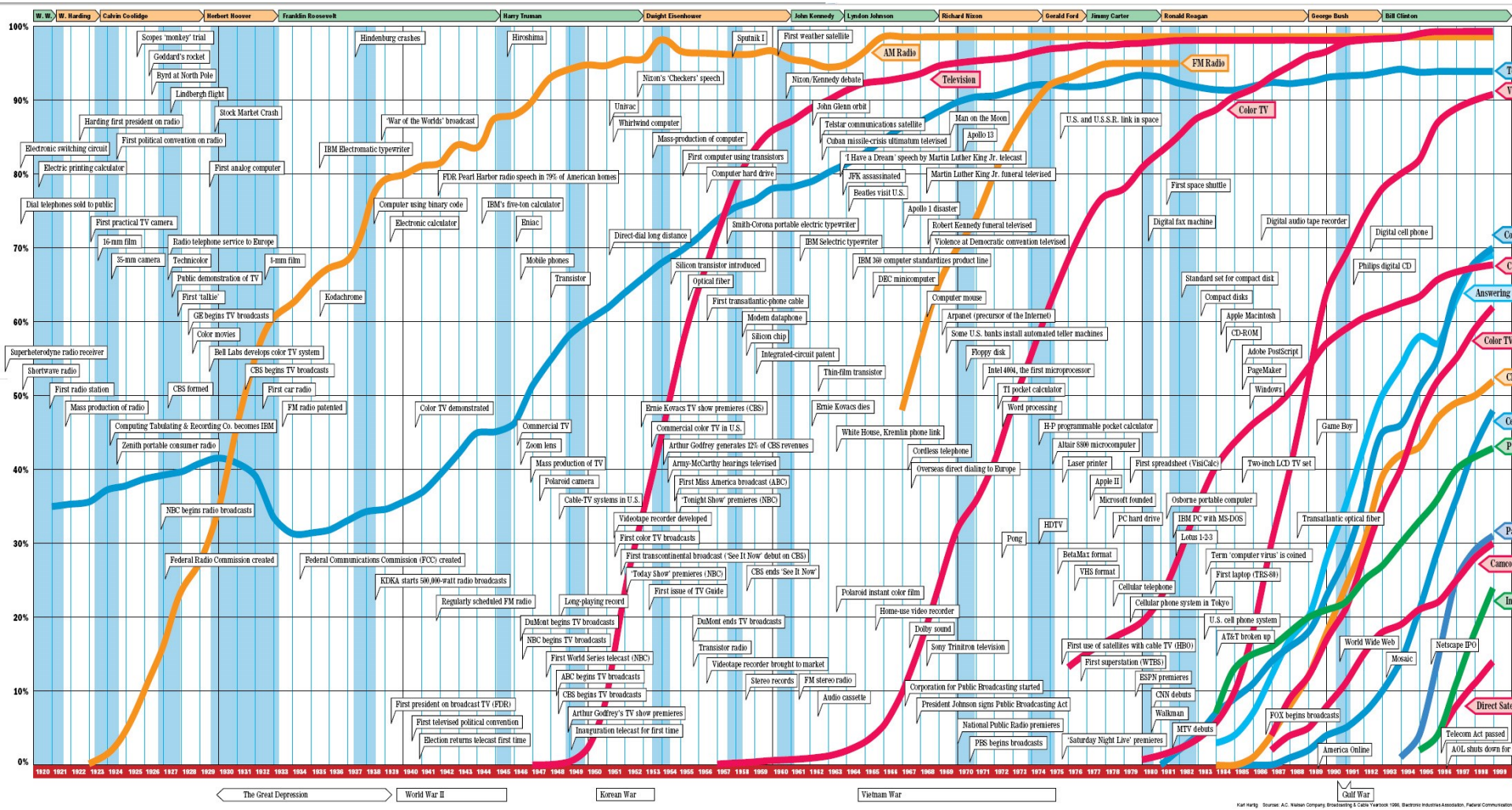


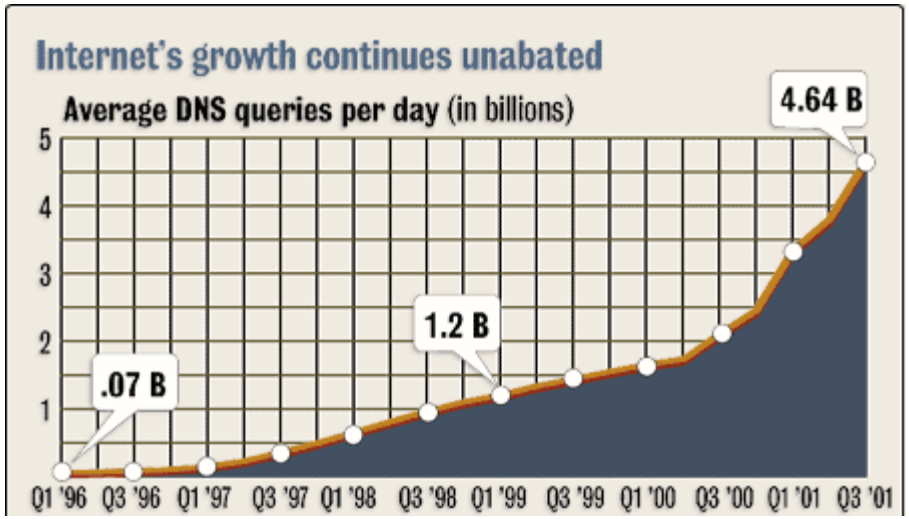
Aug-08

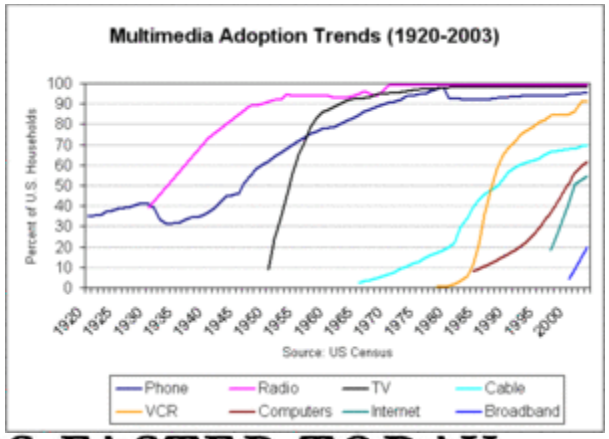
Apr-09

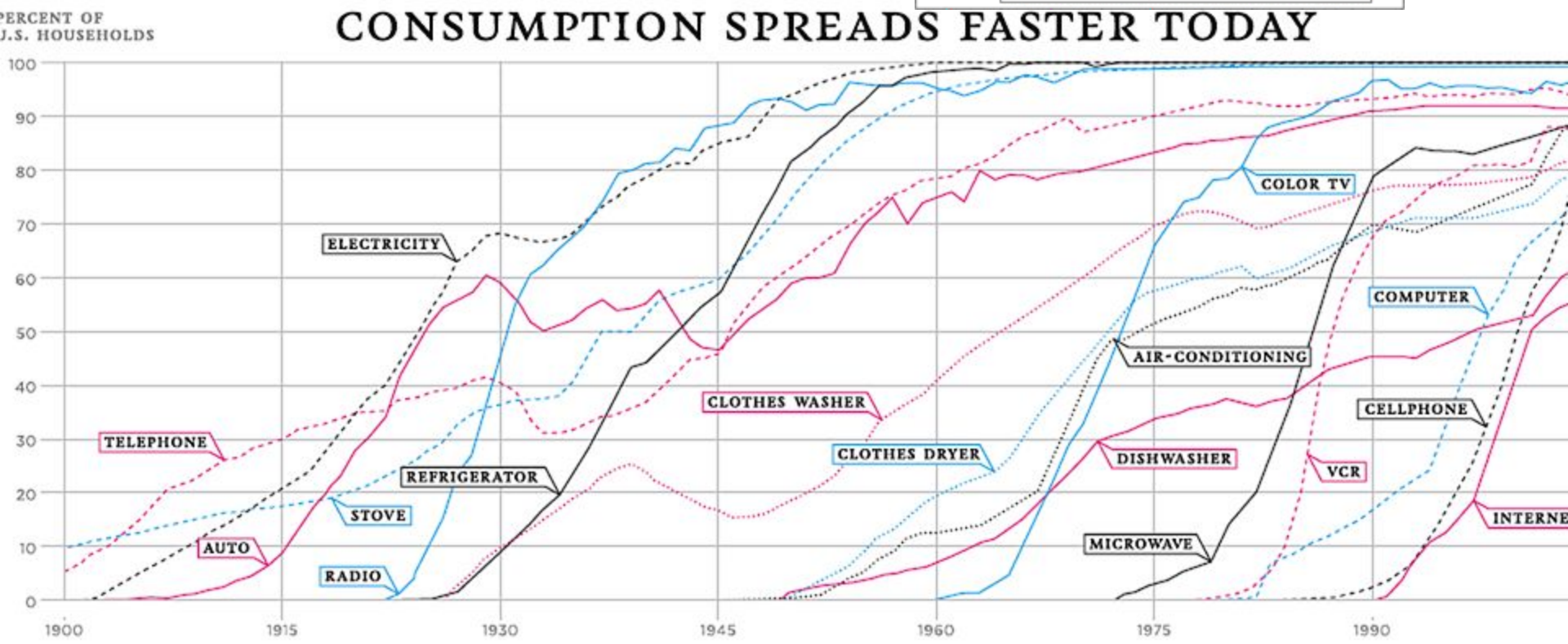
For Next Time

- Add yourself to mailing list
  - We'll send out a key email tomorrow
  - Be sure to get it
- Think about project
  - Form a group (3-4 people)


W. W.
W. Harding
Calvin Coolidge
Herbert Hoover
Franklin Roosevelt
Harry Truman
Dwight Eisenhower
John Kennedy
Lyndon Johnson
Richard Nixon
Gerald Ford
Jimmy Carter
Ronald Reagan
George Bush
Bill Clinton
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
AM Radio
FM Radio
Television
Color TV
Scopes 'monkey' trial
Goddard's rocket
Byrd at North Pole
Lindbergh flight
Harding first president on radio
Stock Market Crash
First political convention on radio
Electronic switching circuit
Electric printing calculator
First analog computer
Dial telephones sold to public
First practical TV camera
16-mm film
35-mm camera
Radio telephone service to Europe
Technicolor
Public demonstration of TV
8-mm film
First 'talkie'
GE begins TV broadcasts
Color movies
Bell Labs develops color TV system
Superheterodyne radio receiver
Shortwave radio
CBS begins TV broadcasts
First radio station
CBS formed
First car radio
Mass production of radio
FM radio patented
Computing Tabulating & Recording Co. becomes IBM
Zenith portable consumer radio
NBC begins radio broadcasts
Federal Radio Commission created
Federal Communications Commission (FCC) created
KDKA starts 500,000-watt radio broadcasts
Regularly scheduled FM radio
Hindenburg crashes
'War of the Worlds' broadcast
IBM Electromatic typewriter
FDR: Pearl Harbor radio speech in 79% of American homes
Computer using binary code
Electronic calculator
IBM's five-ton calculator
Kodachrome
Hiroshima
Eniac
Mobile phones
Transistor
Color TV demonstrated
Commercial TV
Zoom lens
Mass production of TV
Polaroid camera
Cable-TV systems in U.S.
Long-playing record
DuMont begins TV broadcasts
NBC begins TV broadcasts
First World Series telecast (NBC)
ABC begins TV broadcasts
CBS begins TV broadcasts
First president on broadcast TV (FDR)
First televised political convention
Election returns telecast first time
Arthur Godfrey's TV show premieres
Inauguration telecast for first time
Sputnik I
First weather satellite
Nixon's 'Checkers' speech
Univac
Whirlwind computer
Mass-production of computer
First computer using transistors
Computer hard drive
Direct-dial long distance
Silicon transistor introduced
Optical fiber
First transatlantic-phone cable
Modem dataphone
Silicon chip
Integrated-circuit patent
Thin-film transistor
Ernie Kovacs TV show premieres (CBS)
Commercial color TV in U.S.
Arthur Godfrey generates 12% of CBS revenues
Army-McCarthy hearings televised
First Miss America broadcast (ABC)
'Tonight Show' premieres (NBC)
Videotape recorder developed
First color TV broadcasts
First transcontinental broadcast ('See It Now' debut on CBS)
'Today Show' premieres (NBC)
First issue of TV Guide
DuMont ends TV broadcasts
Transistor radio
Videotape recorder brought to market
Stereo records
FM stereo radio
Audio cassette
Nixon/Kennedy debate
John Glenn orbit
Telstar communications satellite
Cuban missile-crisis ultimatum televised
'I Have a Dream' speech by Martin Luther King Jr. telecast
JFK assassinated
Beatles visit U.S.
Martin Luther King Jr. funeral televised
Apollo 1 disaster
Smith-Corona portable electric typewriter
IBM Selectric typewriter
Robert Kennedy funeral televised
Violence at Democratic convention televised
IBM 360 computer standardizes product line
DEC minicomputer
Computer mouse
Arpanet (precursor of the Internet)
Some U.S. banks install automated teller machines
Floppy disk
Intel 4004, the first microprocessor
TI pocket calculator
Word processing
Ernie Kovacs dies
White House, Kremlin phone link
Cordless telephone
Overseas direct dialing to Europe
Polaroid instant color film
Home-use video recorder
Dolby sound
Sony Trinitron television
Corporation for Public Broadcasting started
President Johnson signs Public Broadcasting Act
National Public Radio premieres
PBS begins broadcasts
Man on the Moon
Apollo 13
U.S. and U.S.S.R. link in space
First space shuttle
Digital fax machine
Standard set for compact disk
Compact disks
Apple Macintosh
CD-ROM
Adobe PostScript
PageMaker
Windows
H-P programmable pocket calculator
Altair 8800 microcomputer
Laser printer
Apple II
Microsoft founded
PC hard drive
Osborne portable computer
IBM PC with MS-DOS
Lotus 1-2-3
HDTV
Pong
BetaMax format
VHS format
Cellular telephone
Cellular phone system in Tokyo
First use of satellites with cable TV (HBO)
First superstation (WTBS)
ESPN premieres
CNN debuts
Walkman
MTV debuts
'Saturday Night Live' premieres
First spreadsheet (VisiCalc)
Two-inch LCD TV set
Game Boy
Transatlantic optical fiber
Term 'computer virus' is coined
First laptop (TRS-80)
U.S. cell phone system
AT&T broken up
World Wide Web
Mosaic
Netscape IPO
FOX begins broadcasts
America Online
Digital audio tape recorder
Digital cell phone
Philips digital CD
Answering
Direct Sate
Telecom Act passed
AOL shuts down for
The Great Depression
World War II
Korean War
Vietnam War
Gulf War
1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999
Karl Hartig Sources: A.C. Nielson Company, Broadcasting & Cable Yearbook 1996, Electronic Industries Association, Federal Communication


Internet's growth continues unabated
Average DNS queries per day (in billions)
4.64 B
1.2 B
.07 B
5
4
3
2
1
Q1 '96
Q3 '96
Q1 '97
Q3 '97
Q1 '98
Q3 '98
Q1 '99
Q3 '99
Q1 '00
Q3 '00
Q1 '01
Q3 '01


Multimedia Adoption Trends (1920-2003)
Percent of U.S. Households
100
90
80
70
60
50
40
30
20
10
0
1920
1925
1930
1935
1940
1945
1950
1955
1960
1965
1970
1975
1980
1985
1990
1995
2000
Source: US Census
Phone
Radio
TV
Cable
VCR
Computers
Internet
Broadband


PERCENT OF
U.S. HOUSEHOLDS
CONSUMPTION SPREADS FASTER TODAY
100
90
80
70
60
50
40
30
20
10
0
1900
1915
1930
1945
1960
1975
1990
ELECTRICITY
TELEPHONE
AUTO
RADIO
REFRIGERATOR
STOVE
CLOTHES WASHER
CLOTHES DRYER
COLOR TV
AIR-CONDITIONING
DISHWASHER
MICROWAVE
COMPUTER
CELLPHONE
VCR
INTERNE

TODO

- Include section difference between information (unstructured) and data (structured)

KEY Topics

- Data and Information
- Precision and Recall
- Vector Space Model
- Document Similarity

Information Retrieval

- Computers are very good at storing information.
  - Eg The web
- But information is useless unless you can access (retrieve) it.
  - Eg The web
- Why is this a problem? What's wrong with
  select * from information where text like '%dog%'
- Problem lies in distinction between information and data

Data *vs* Information

- The terms *data* and *information* are used in various ways, but crucial difference
  - Data is *structured*
  - Information is *unstructured*

Exercise: Information Retrieval or Data Retrieval?

- Using the library catalogue system to find how many books you have out
- Using the library catalogue system to find a good introduction to intelligent systems
- Finding documents on your hard drive that mention dogs
- Finding pictures on your hard drive that have dogs in

Information Overload

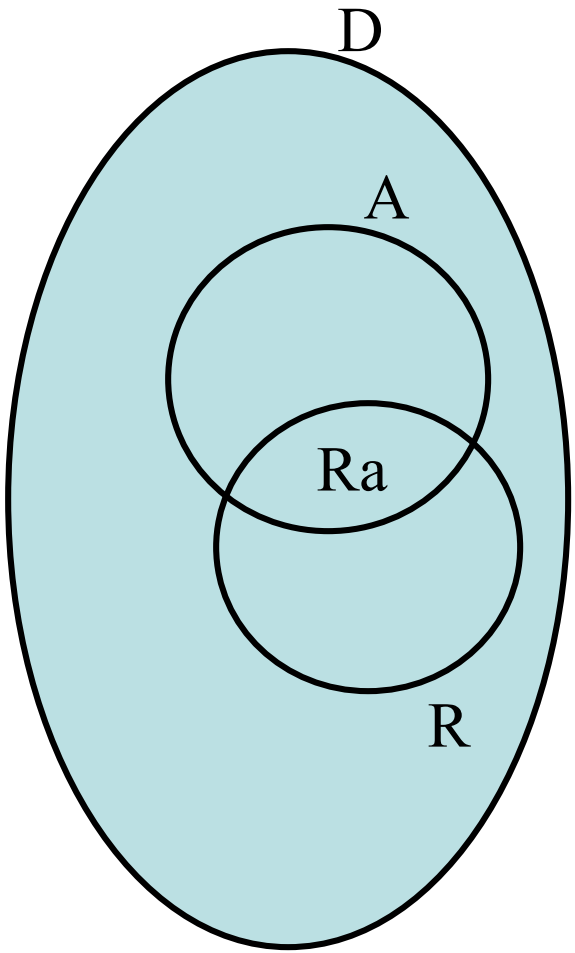
- Another problem: collections of data (especially the web) are now so large, that the ‘right’ information is often swamped
  - http://www.google.com/search?q=dog
    = 36 million pages
- How do we find the most *relevant* (or *best for our needs*)
- Also a problem of semantics

Precision *vs* Recall

- A way of formally measuring the performance of an information retrieval system using two measures
- **Precision** = proportion of retrieved documents that are relevant
- **Recall** = proportion of relevant documents that are retrieved

Precision and Recall Defined

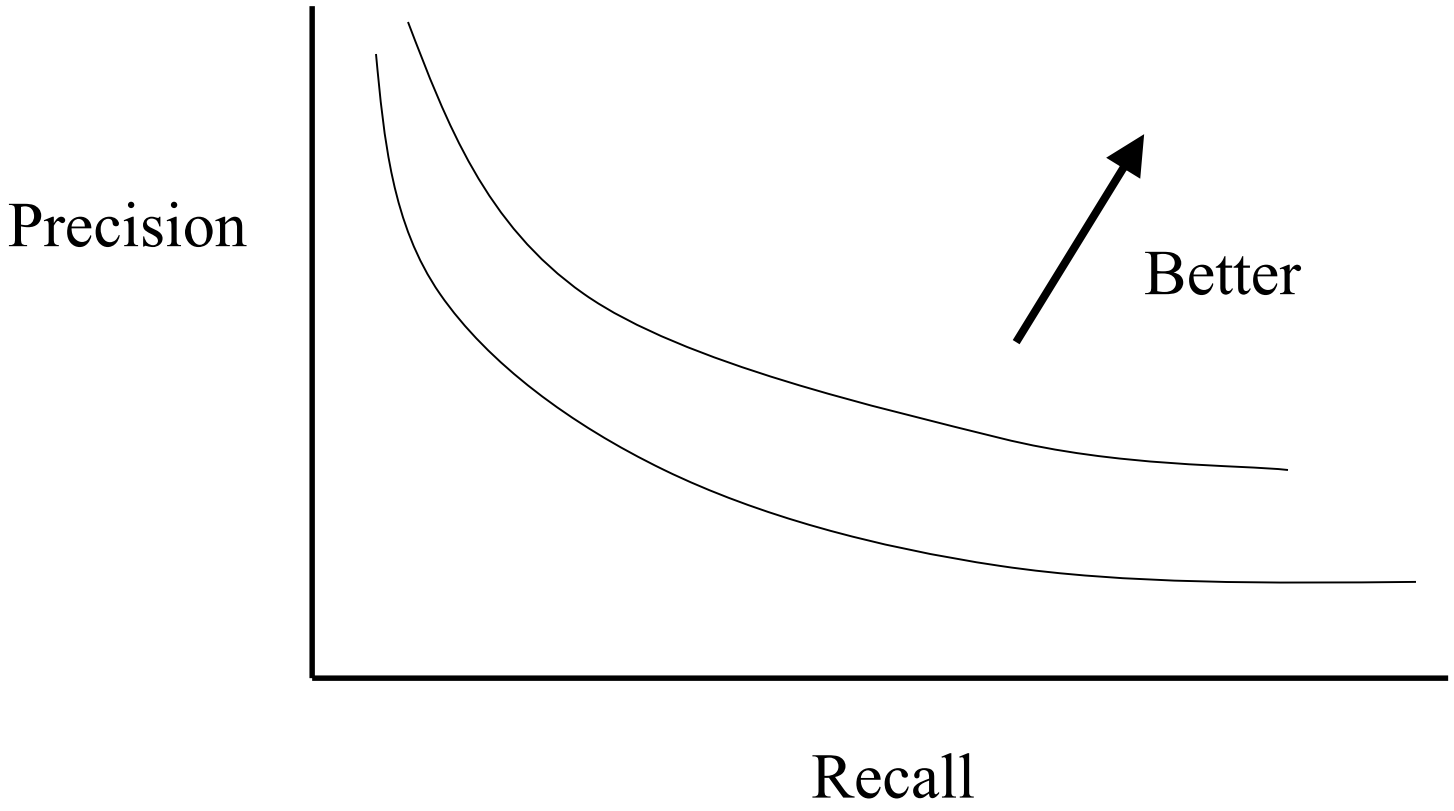
- Suppose we have:
  - D = set of documents
  - R $\subset$ D = relevant documents
  - A $\subset$ D = retrieved documents
  - Ra $\subset$ D = relevant documents retrieved
- *Recall* = |Ra| / |R|
- *Precision* = |Ra| / |A|



Precision *vs* Recall

- More documents retrieved
  - = higher recall, lower precision
  - Hard to find right information in retrieved documents
- Less documents retrieved
  - = lower recall, higher precision
  - Might miss crucial information

Precision / Recall Graphs



Precision or Recall

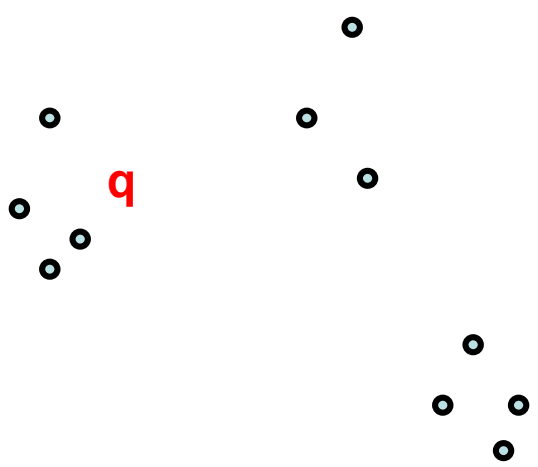
- Different tasks require precision or recall
- What is most important for the following?:
  - Finding books about information retrieval in the library
  - Finding information about information retrieval on the web
  - Finding emails related to terrorist activity

The Vector Space Model

- A very successful way of thinking about documents for document retrieval
- AKA “word bag”: ignores word order and just counts instances of words in a document
- Introduced by Salton in SMART system, 60s
- Allows us to measure the *similarity* between a query and a document (or between two documents)
  - Which helps rank search results
- Doesn’t solve the problem of search on its own, but provides a mathematical framework for handling them

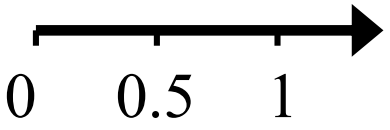
Documents as Vectors

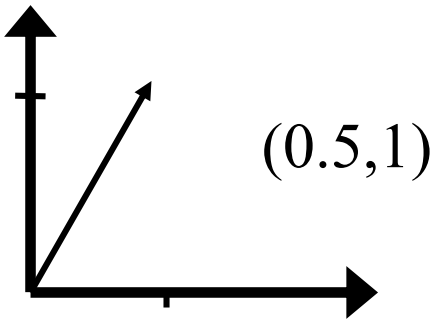
- Once documents are represented as vectors can:
  - Classify
  - Cluster
  - Find documents closest to a query (ie search)

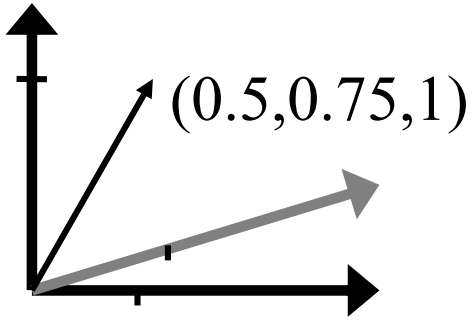


Vector Spaces

- A point in a vector space can be described by a sequence of numbers:
- 1D Vector Space = a line
- Each point described using a single number
- 2D Vector Space = a plane
- Each point described using two numbers
- 3d Vector Space = ‘normal’ space
- Each point described using three numbers







Higher Dimensional Vector Spaces

- We can't really visualize higher dimensional vector spaces, but the same principal applies:
  - 1 number ('ordinate' or 'component') per dimension
- 4D vector space
  - $\mathbf{A} = (a_1, a_2, a_3, a_4)$
- N-Dimensional vector space
  - $\mathbf{A} = (a_1, a_2, a_3, \ldots, a_N)$

Documents as Vectors

- N-dimensional space
  - N is the number of *index terms* in the document collection (large!)
  - Each document / query is a vector in this space
- Doc_1 = “A dog is a big animal”
- Doc_2 = “A cat is a furry animal”
- K = (a, is, dog, cat, animal, big, furry)
  - $\mathbf{d_1}$ = (2, 1, 1, 0, 1, 1, 0)
  - $\mathbf{d_2}$ = (2, 1, 0, 1, 1, 0, 1)
- Example query = “big animal”
  - **q** = (0, 0, 0, 0, 0, 1, 1, 0)

Stemming and Stopping

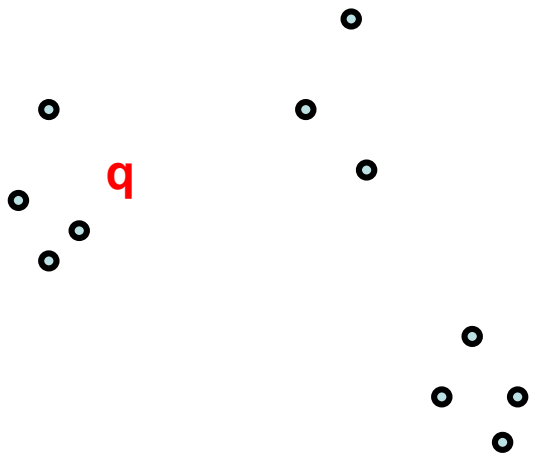
- **Problem:** Too many words (index terms)!
- *A, is, an, the,* etc – so common, contain little information
  - So ignore ‘stop’ words completely
- *Jump, jumps, jumped,* etc – different version of the same word *jump*
  - So just record the stemmed version
  - Eg Porter Stemming algorithm
  - http://maya.cs.depaul.edu/~classes/ds575/porter.html
  - http://www.tartarus.org/~martin/PorterStemmer/

Stemmed and Stopped Space

- Doc_1 = “A dog is a big animal”
- Doc_2 = “A cat is a furry animal”
- K = (dog, cat, animal, big, furry)
  - $\mathbf{d_1}$ = (1, 0, 1, 1, 0)
  - $\mathbf{d_2}$ = (0, 1, 1, 0, 1)
- Example query = “big animal”
  - **q** = (0, 0, 1, 1, 0)

Similarity in Vector Space

- **Question:** How do we define similarity between documents / queries?
- **First Answer**: Why not use Euclidean distance?
  - Similarity is inverse of distance
  - *ie* small Euclidean distance = very similar documents



Euclidean Document Distances

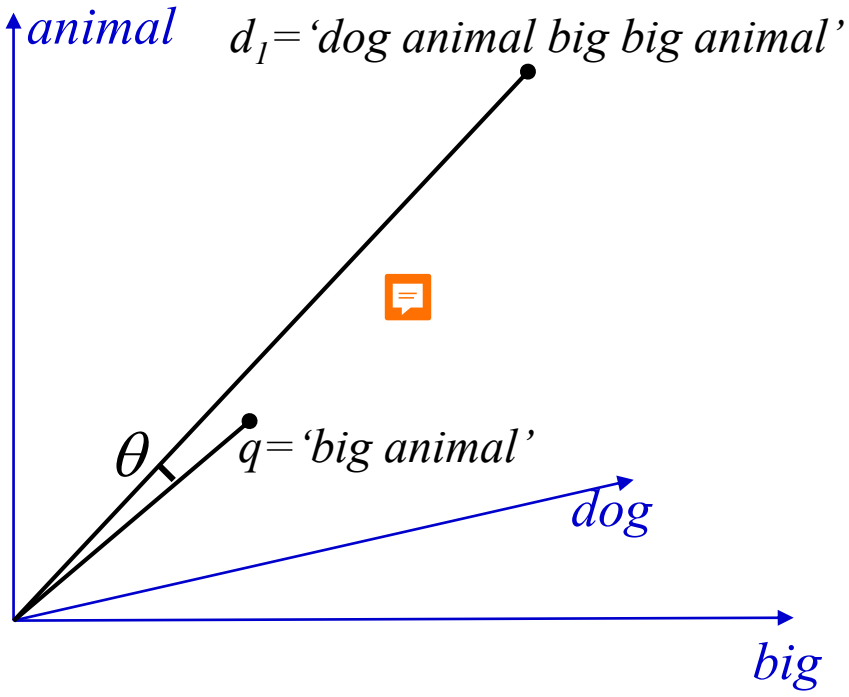
- Doc_1 = “A dog is a big animal”
- Doc_2 = “A cat is a furry animal”
- K = (dog, cat, animal, big, furry)
  - $\mathbf{d_1}$ = (1, 0, 1, 1, 0)
  - $\mathbf{d_2}$ = (0, 1, 1, 0, 1)
- Example query = “big animal”
  - **q** = (0, 0, 1, 1, 0)
- $D(\mathbf{q},\mathbf{d_1}) = \sqrt{(1-0)^2+ (0-0)^2+ (1-1)^2+ (1-1)^2 + (0-0)^2} = 1$
- $D(\mathbf{q},\mathbf{d_2}) = \sqrt{(0-0)^2+ (1-0)^2+ (1-1)^2+ (0-1)^2 + (1-0)^2} = 1.73$
- Therefore $\mathbf{d_1}$ is the best match (most similar, least distance) to **q**

Euclidean Document Distances

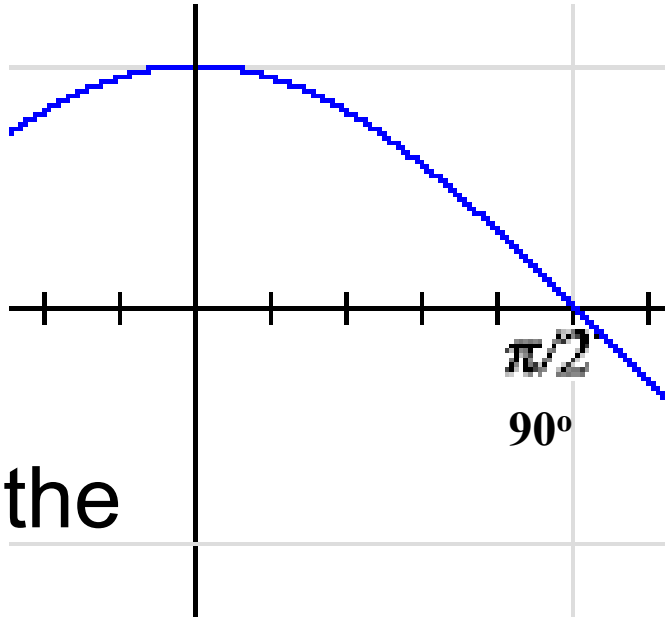
- *But,* now suppose we have:
  - Doc_1 = “A dog is a big animal. A dog is a *big, big* animal. ”
  - Doc_2 = “A cat is a furry animal”
  - Example query = “big animal”
- K = (dog, cat, animal, big, furry)
  - $\mathbf{d_1}$ = (2, 0, 2, 3, 0)
  - $\mathbf{d_2}$ = (0, 1, 1, 0, 1)
  - $\mathbf{q}$ = (0, 0, 1, 1, 0)
- $D(\mathbf{q},\mathbf{d_1}) = \sqrt{(2-0)^2+ (0-0)^2+ (2-1)^2+ (3-1)^2 + (0-0)^2} = 3$
- $D(\mathbf{q},\mathbf{d_2}) = \sqrt{(0-0)^2+ (1-0)^2+ (1-1)^2+ (0-1)^2 + (1-0)^2} = 1.73$
- Now Doc_2 is the best match!!?

Similarity between document vectors

- Size of document (vector) shouldn't matter.
- Only interested in the difference in direction
- Smaller angle = greater similarity



Angles and Cosines



- All weights are positive so $0 < \cos \theta < 1$
- Smaller the angle, greater the cosine, and greater the similarity between documents
- Salton originally proposed this cosine measure

Finding Cosines

- How do we find the cosine of the angle between vectors?
- *Normalised dot product*
  $= \mathbf{A} \cdot \mathbf{B} \,/\, |\mathbf{A}||\mathbf{B}|$
  where:
  - $\mathbf{A} = (a_1, a_2, \ldots a_N)$, $\mathbf{B} = (b_1, b_2, \ldots b_N)$
  - $\mathbf{A} \cdot \mathbf{B} = (a_1.b_1)+(a_2.b_2)+\ldots+(a_N.b_N)$
  - $|\mathbf{A}| = \sqrt{(a_1^2+a_2^2+\ldots+a_N^{2)}}$

Cosine Measure, example

- Document Set:
  - Doc_1 = “A dog is a big animal. A dog is a big, big animal.”
  - Doc_2 = “A cat is a furry animal”
  - Query = “big animal”
- K = (dog, cat, animal, big, furry)
  - $\mathbf{d_1}$ = (2, 0, 2, 3, 0) $|\mathbf{d_1}| = \sqrt{4+0+4+9+0} = 4.12$
  - $\mathbf{d_2}$ = (0, 1, 1, 0, 1) $|\mathbf{d_2}| = \sqrt{0+1+1+0+1} = 1.73$
  - $\mathbf{q}$ = (0, 0, 1, 1, 0) $|\mathbf{q}| = \sqrt{0+0+1+1+0} = 1.41$

Cosine Measure Example

- Cos(q,Doc1) = $\mathbf{q}.\mathbf{d}_1 / |\mathbf{q}||\mathbf{d}_1|$
  = (2x0) + (0x0) + (2x1) + (3x1) + (0x0) / (1.41x4.12)
  = 0.857
- Cos(q,Doc1) = $\mathbf{q}.\mathbf{d}_2 / |\mathbf{q}||\mathbf{d}_2|$
  = (0x0) + (1x0) + (1x1) + (0x1) + (1x0) / (1.41x1.73)
  = 0.408
- So Doc1 is more relevant to q than Doc2!!!

In Machine Learning

- What really matters? How it can be done?
- Demo as well.

Similarity Metrics

How this will Happen?

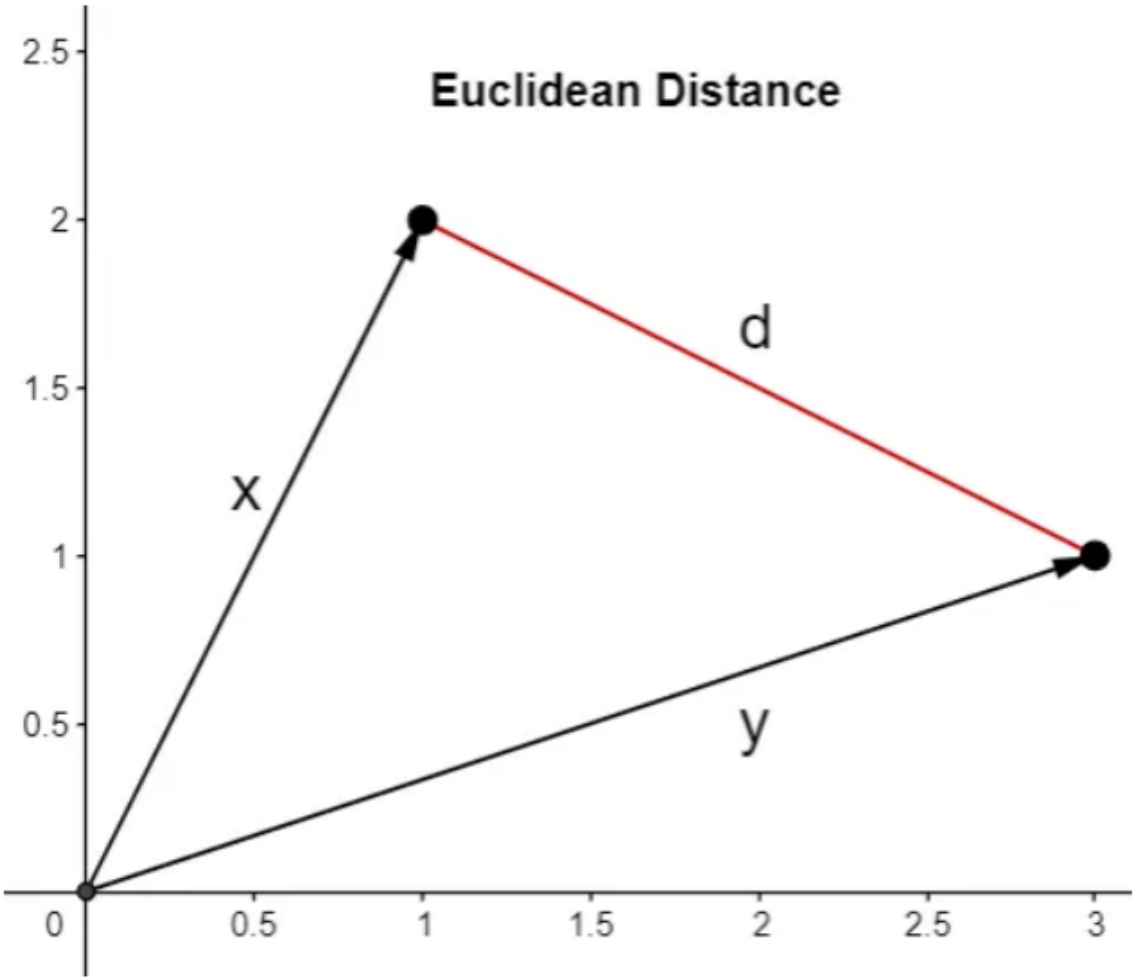
- Evaluating the similarity between documents is a common step in NLP. It plays an integral role in many microservices with functions like information retrieval and translation.
- Take a moment to ask yourself: how does a machine compare text in the first place?

How this will Happen?

- If you want to harness these techniques for yourself, you'll need to understand a bit of the math that goes behind the execution of text comparison.
- Similarity metrics, albeit naïve, are a great way to introduce yourself to the concept of comparing documents. Here, we explore two of the more well-known similarity metrics: Euclidean distance and cosine similarity.

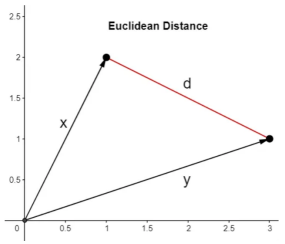
Euclidean distance

- Remember that in NLP, bodies of text are represented as vectors.
- So, from a mathematical perspective, what would be the most suitable way to compare vectors?
- Intuitively, comparing the distance between the two vectors seems like the most logical approach.
- The metric that measures the distance between two vectors is the **Euclidean distance.**



$$Eucliden\ Distance = d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

ED: Diagram



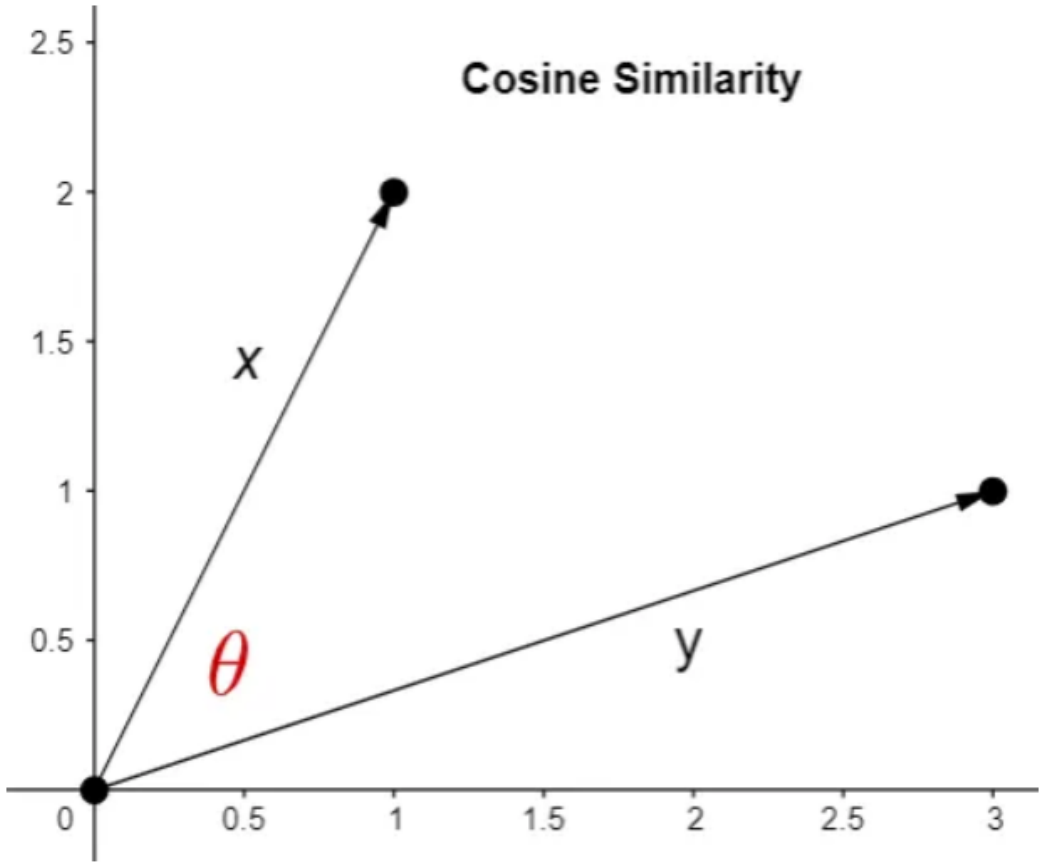
- The distance between vectors x and y are dependent on the magnitude of those vectors. In this case, an increase in magnitude of vector x or y would result in a larger Euclidean distance.
- In terms of NLP, this means that a larger body of text, which has more words in terms of variety and frequency, will have a much greater magnitude than a smaller body text even if they share the same topic.
- This can be a problem since people comparing documents are more interested in the subject of the documents than the amount of text.

Cosine Similarity

- Documents, represented as vectors, can be compared by evaluating the angle between two vectors.
- The metric that considers the angle of two vectors is the **cosine similarity metric**.

$$Cosine\ Similarity = cos(\theta) = \frac{x \cdot y}{\|x\|\ \|y\|}$$

- The cosine similarity value can range from 0 to 1, with 0 representing the lowest similarity and 1 representing the highest similarity.



The angle between vectors x and y will remain the same regardless of how much vectors x and y change in magnitude.

This means that the cosine similarity considers the angle between two vectors without examining the magnitude of the vectors. This eliminates the problem brought by the Euclidean distance metric.

Python –Cosine Similiraty

Consider the following 3 sentences:

- Sentence 1: I like eating ice cream on a hot summer day.
- Sentence 2: Only boring people do not like eating ice cream.
- Sentence 3: I do not like going out during the summer since it is so hot.

Python -src

- We can evaluate the similarity between these sentences by deriving the cosine of the angle between these sentences' vector representations

Sklearn module’

- Thanks to Python Sklearn module ☺
- First, let’s create a function that can compute the cosine similarity with two given texts. This function will convert the given sentences to vectors with the TF-IDF algorithm and compute the cosine similarity of the generated vectors.

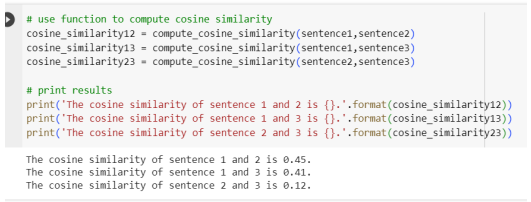
Class Activity

- Open GoogleColab

```
# use function to compute cosine similarity
cosine_similarity12 = compute_cosine_similarity(sentence1,sentence2)
cosine_similarity13 = compute_cosine_similarity(sentence1,sentence3)
cosine_similarity23 = compute_cosine_similarity(sentence2,sentence3)

# print results
print('The cosine similarity of sentence 1 and 2 is {}.'.format(cosine_similarity12))
print('The cosine similarity of sentence 1 and 3 is {}.'.format(cosine_similarity13))
print('The cosine similarity of sentence 2 and 3 is {}.'.format(cosine_similarity23))
```

```
The cosine similarity of sentence 1 and 2 is 0.45.
The cosine similarity of sentence 1 and 3 is 0.41.
The cosine similarity of sentence 2 and 3 is 0.12.
```



```
# use function to compute cosine similarity
cosine_similarity12 = compute_cosine_similarity(sentence1,sentence2)
cosine_similarity13 = compute_cosine_similarity(sentence1,sentence3)
cosine_similarity23 = compute_cosine_similarity(sentence2,sentence3)

# print results
print('The cosine similarity of sentence 1 and 2 is {}.'.format(cosine_similarity12))
print('The cosine similarity of sentence 1 and 3 is {}.'.format(cosine_similarity13))
print('The cosine similarity of sentence 2 and 3 is {}.'.format(cosine_similarity23))
```

```
The cosine similarity of sentence 1 and 2 is 0.45.
The cosine similarity of sentence 1 and 3 is 0.41.
The cosine similarity of sentence 2 and 3 is 0.12.
```

Result Analysis

- Based on the results, sentences 1 and 2 and sentences 1 and 3 have the highest similarities, while sentences 2 and 3 have the lowest similarity.
- This makes sense as sentences 1 and 2 both mention ‘eating ice cream’ while sentences 1 and 3 both mention a ‘hot summer’. On the other hand, sentences 2 and 3 don’t really share any topics, thus leading to a low cosine similarity score

To be Cont’d..

- Next week: