Internet-Scale Data Analysis

Peter Norvig
Google

define: Internet Scale

- Data center scale

- Warehouse scale

Types of Data

- Feature vectors: Data =

$$\{x_{1,1},\ x_{1,2},\ \dots,\ x_{1,n}$$

$$x_{1,1},\ x_{1,2},\ \dots,\ x_{1,n}$$

$$\dots$$

$$x_{m,1},\ x_{m,2},\ \dots,\ x_{m,n}\}$$

- Also, data = graphs, images, text, ...

Internet Scale Data

- m = Billions to trillions of examples
- n = Millions to billions of features

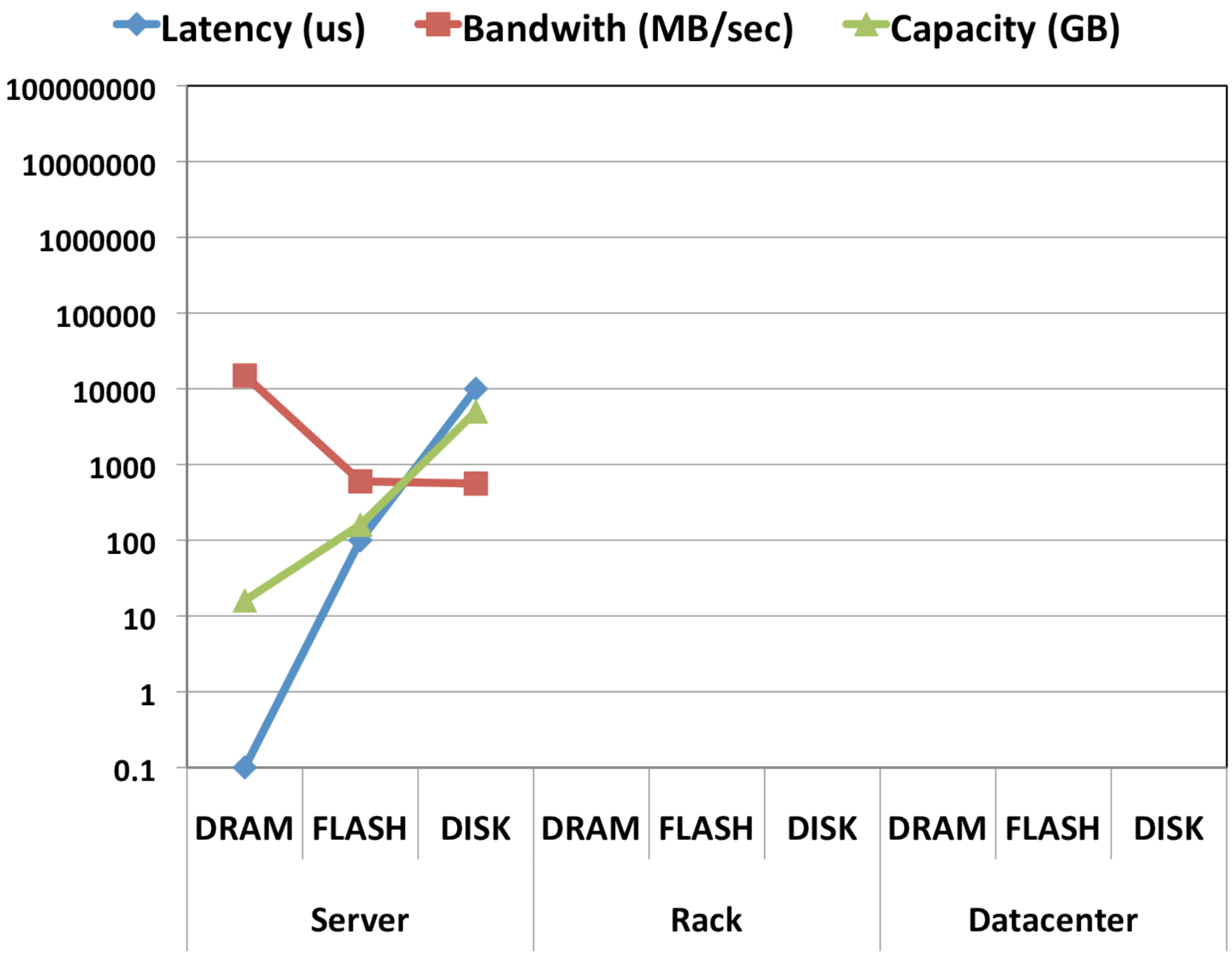
- Hundreds to thousands of CPUs

- Data is noisy
- Data streams in
- Unpredictable query demand

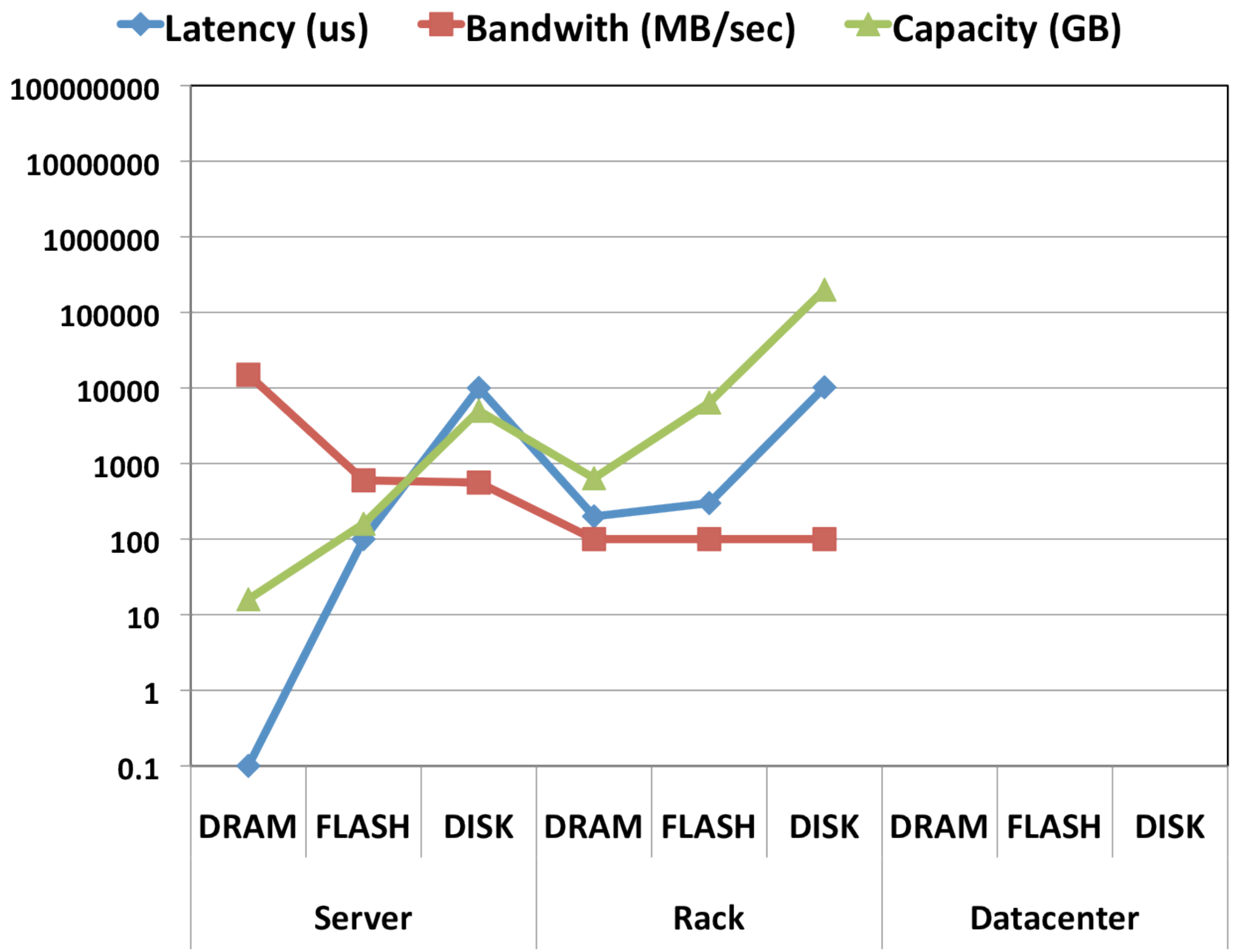
Sample Hierarchy

- Server
  16GB DRAM; 160 GB SSD; 5 x 1TB disk
- Rack
  40 servers
  48 port Gigabit Ethernet switch
- Warehouse
  10,000 servers (250 racks)
  2K port Gigabit Ethernet switch

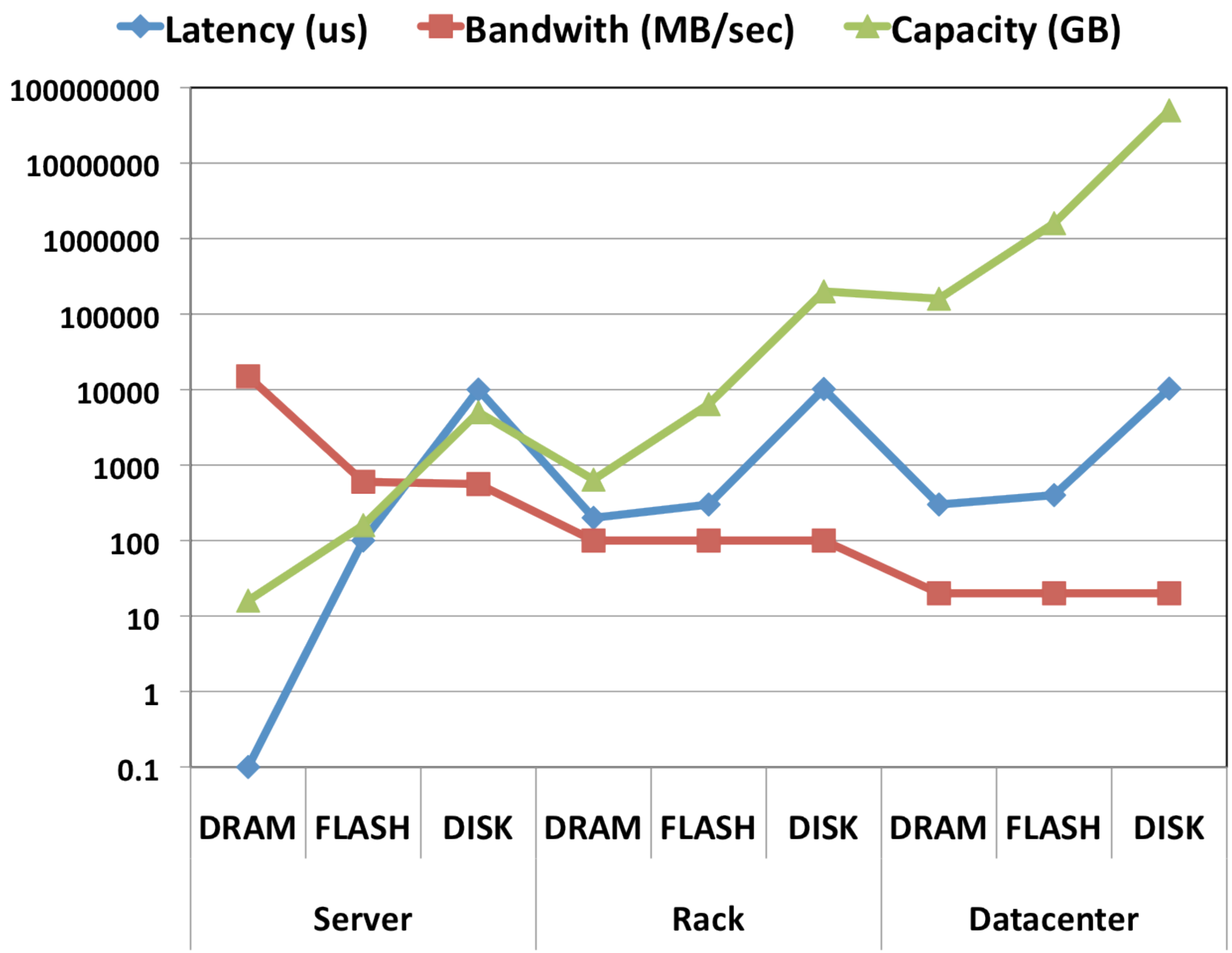
Storage hierarchy – single server



Storage hierarchy – one rack



Storage hierarchy – WSC



Challenges

- New programming models:
  - Parallel; Flash (SSD); GPUs?
- Use energy efficiently
  - Hardware, software, warehouse
- Encode/compress/transmit data well
- Fault Recovery
  - Deal with stragglers
  - Harware/software faults
  - Heavy tail

Distributed Program Design

- Failure is always an option
- Minimize network traffic
- Experiments/back-of-envelope
- Caching and replication
- Minimize average latency
- Minimize variance (long tail) of latency

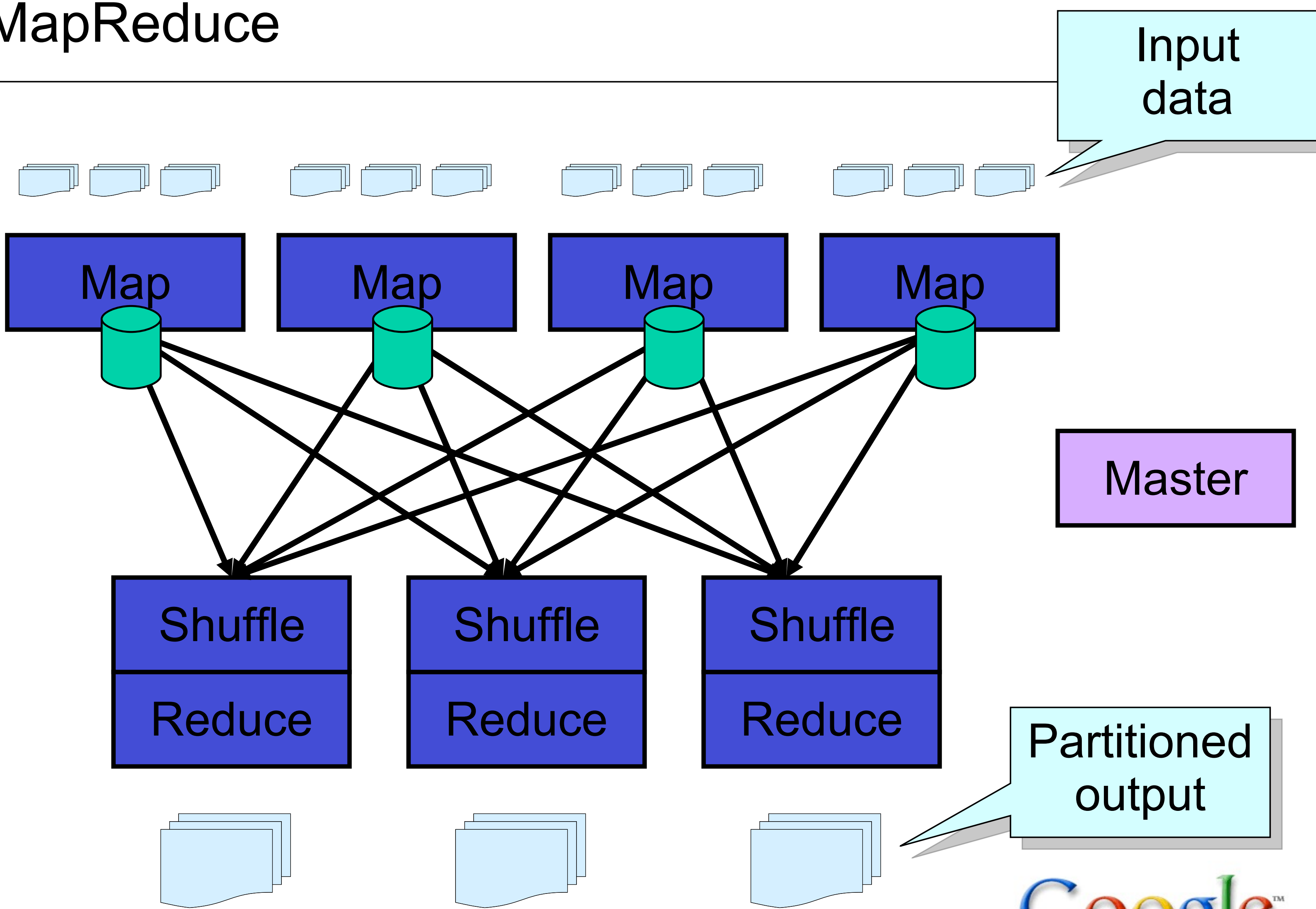
The Eight Fallacies (Peter Deutsch)

- The network is reliable
- Latency is zero
- Bandwidth is infinite
- The network is secure
- Topology doesn't change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

Map-reduce model

- Distributed, stateless computation
- Built-in failure recovery
- Built-in load balancing
- Network and storage optimizations
- Built-in sort of intermediate values
- Various interfaces (file system, etc.)
- Protocol buffers for structured data

MapReduce



Input data
Map
Map
Map
Map
Master
Shuffle
Reduce
Shuffle
Reduce
Shuffle
Reduce
Partitioned output

MapReduce: Granularity

Fine granularity tasks: many more map tasks than machines

- Minimizes time for fault recovery
- Can pipeline shuffling with map execution
- Better dynamic load balancing

Process	Time --------------------->						
User Program	MapReduce()	... wait ...					
Master	Assign tasks to worker machines...						
Worker 1	Map 1	Map 3					
Worker 2	Map 2						
Worker 3	Read 1.1		Read 1.3		Read 1.2	Reduce 1	
Worker 4	Read 2.1	Read 2.2	Read 2.3	Reduce 2			

Mapreduce

```
// word count
map(String key, String value):
    // key: document name
    // value: document contents
    for each word w in value:
      EmitIntermediate(w, "1");

reduce(String key, Iterator values):
    // key: a word
    // values: a list of counts
    int result = 0;
    for each v in values:
      result += ParseInt(v);
    Emit(AsString(result));
```

A Comparison of Approaches to Large-Scale Data Analysis

Andrew Pavlo
Brown University
pavlo@cs.brown.edu

Erik Paulson
University of Wisconsin
epaulson@cs.wisc.edu

Alexander Rasin
Brown University
alexr@cs.brown.edu

Daniel J. Abadi
Yale University
dna@cs.yale.edu

David J. DeWitt
Microsoft Inc.
dewitt@microsoft.com

Samuel Madden
M.I.T. CSAIL
madden@csail.mit.edu

Michael Stonebraker
M.I.T. CSAIL
stonebraker@csail.mit.edu

ABSTRACT

There is currently considerable enthusiasm around the MapReduce (MR) paradigm for large-scale data analysis [17]. Although the basic control flow of this framework has existed in parallel SQL database management systems (DBMS) for over 20 years, some have called MR a dramatically new computing model [8, 17]. In this paper, we describe and compare both paradigms. Furthermore,

model through which users can express relatively sophisticated distributed programs, leading to significant interest in the educational community. For example, IBM and Google have announced plans to make a 1000 processor MapReduce cluster available to teach students distributed programming.

Given this interest in MapReduce, it is natural to ask "Why not use a parallel DBMS instead?" Parallel database systems (which

"A major step backwards"


UR DOING IT
WRONG!1

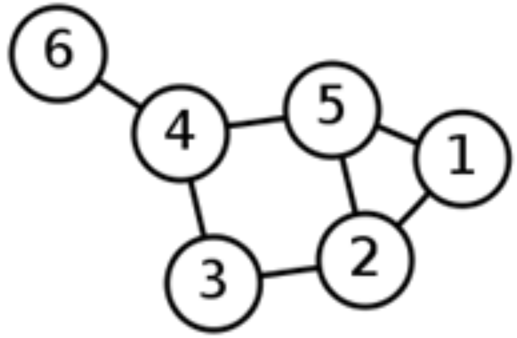
Claims

- MapReduce cannot use indices and implies a full scan of the input data
- MapReduce input and outputs are always simple files in a file system
- MapReduce requires using inefficient textual data formats

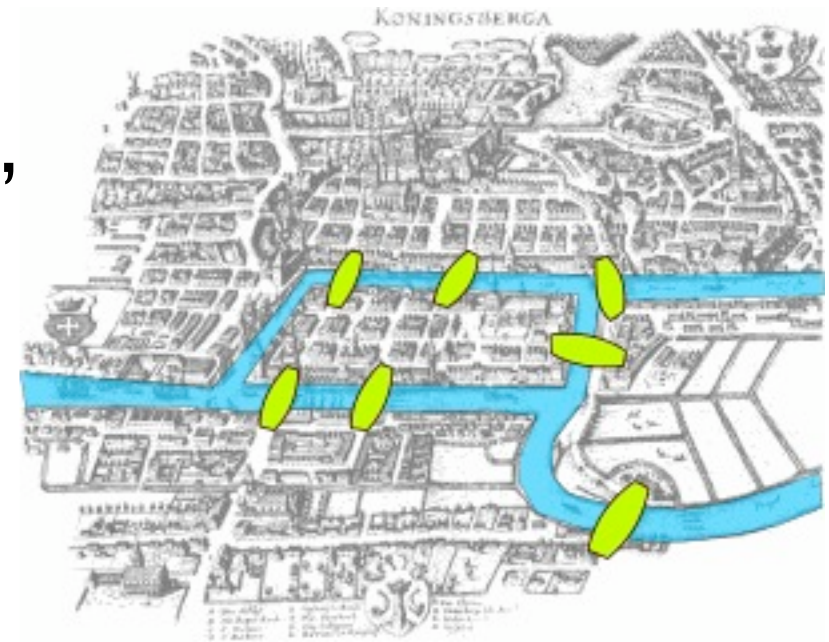
Bigtable

- Sparse, distributed, multi-dimensional sorted map
- Column oriented (roughly, columns for OLAP, rows for OLTP)
- Heavy use of compression
- Has locks, but designed for many queries, not for transactions

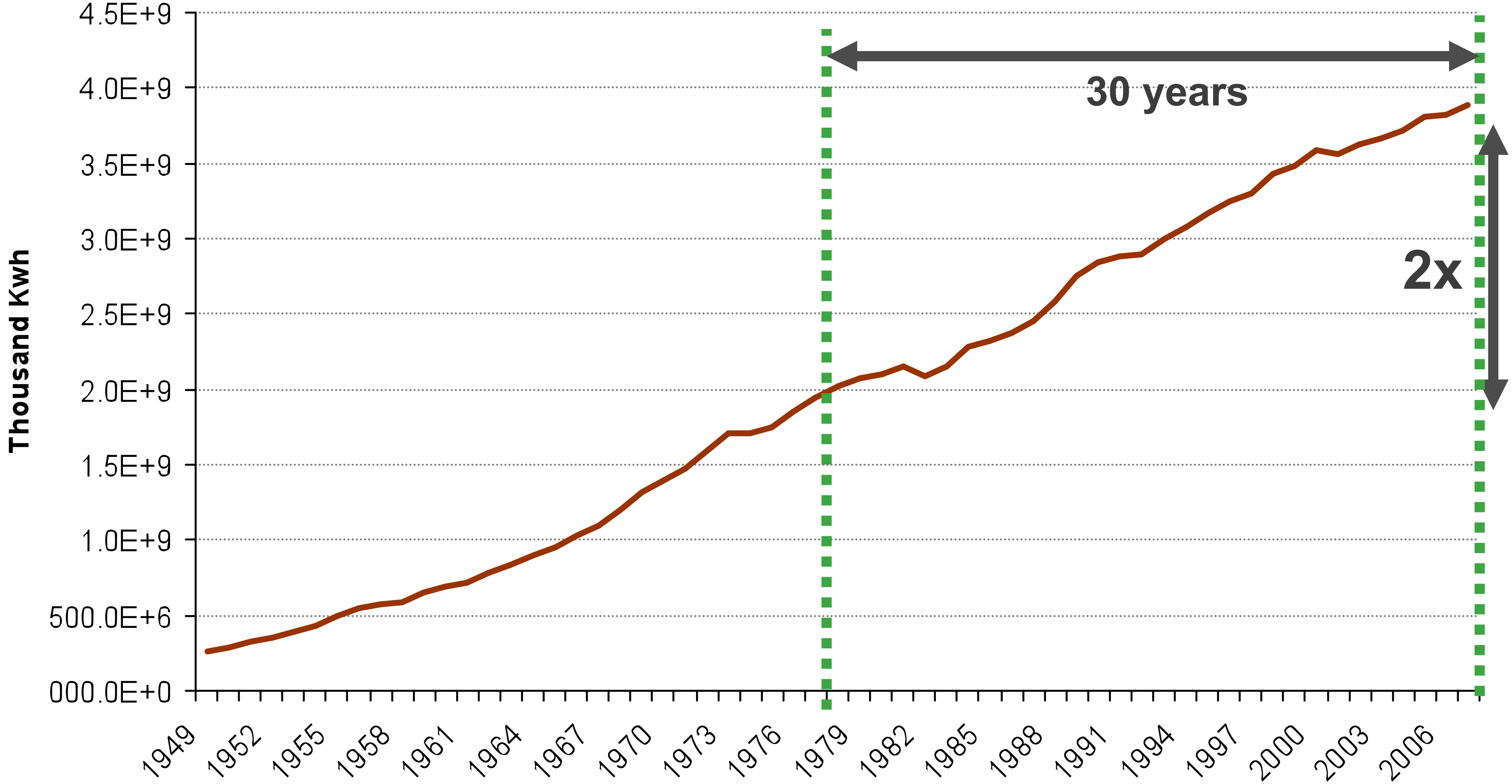
Pregel



- Graph processing
- Bulk synchronous parallel model
- Message passing to vertexes
- Billions of vertexes,
- “Think like a vertex”



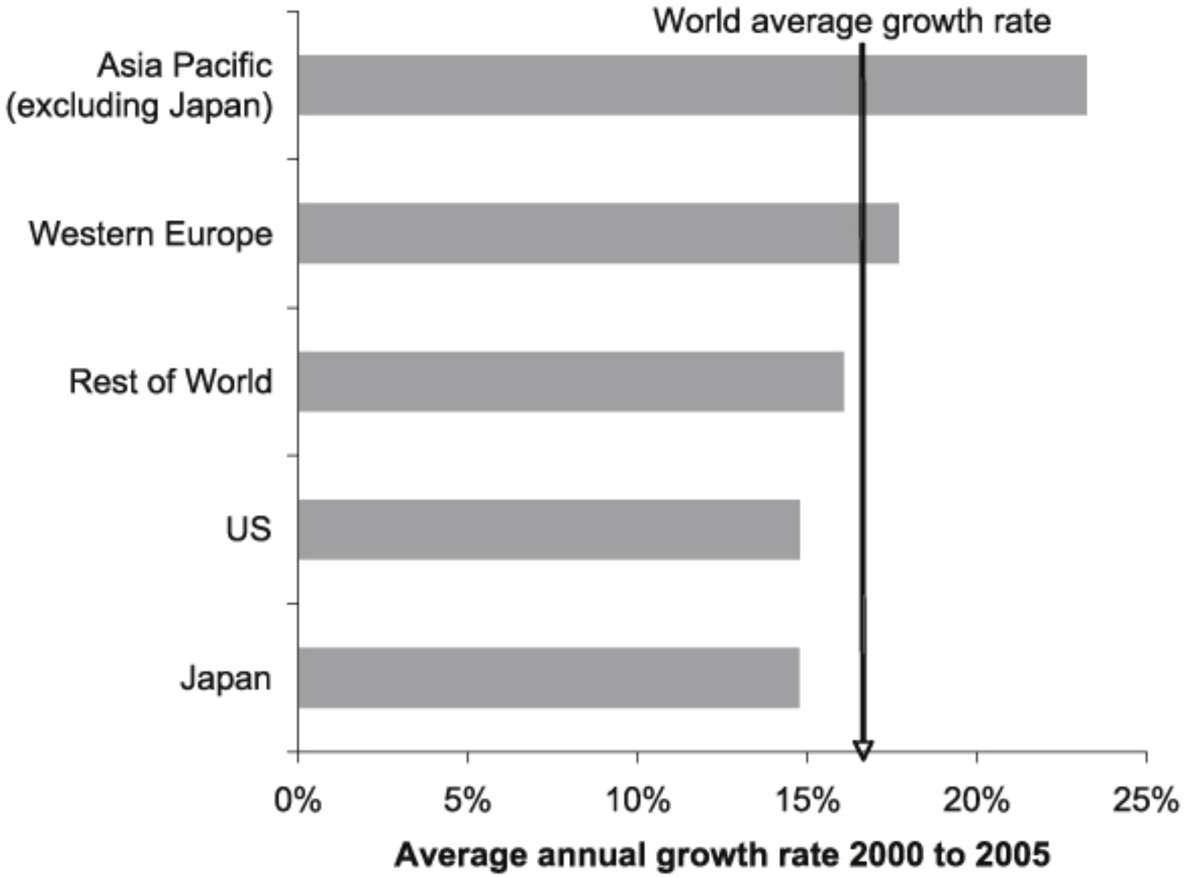
Supply (US Electricity Output)



Doubling every ~30 years

source: http://eia.doe.gov

Demand (aggregate energy usage of servers)



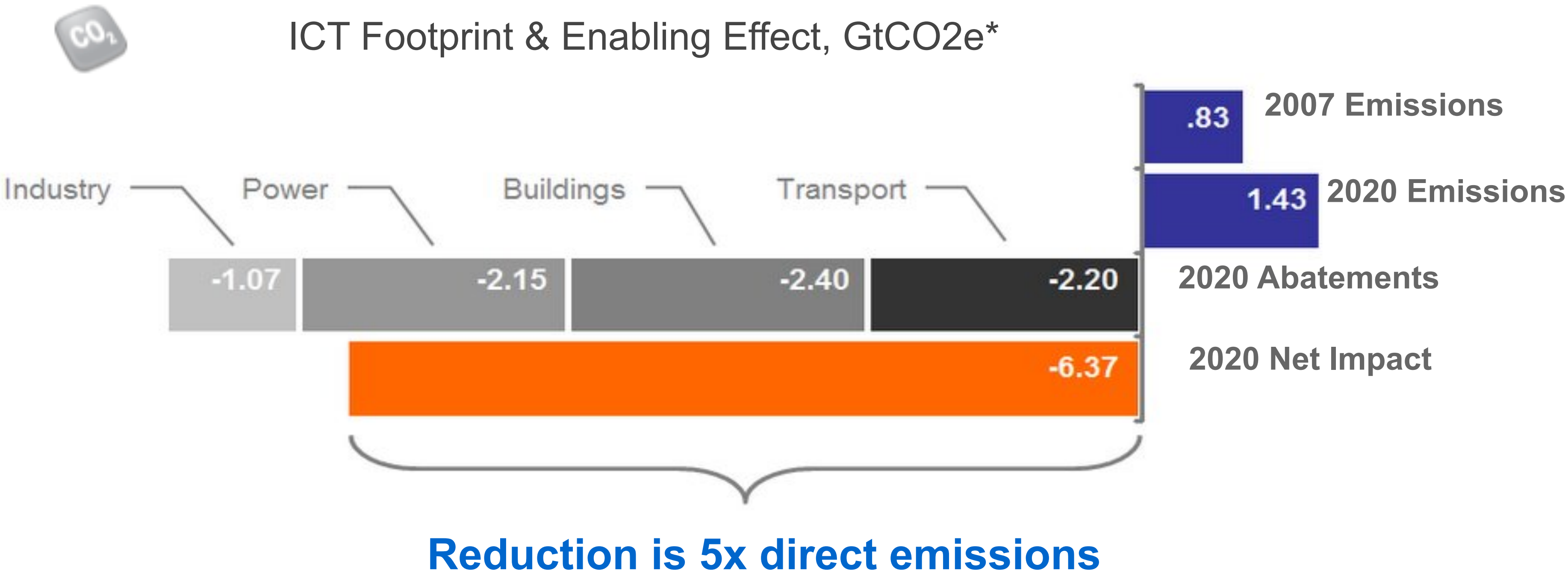
Doubling every 6 – 8 years

source: J. Koomey,"Worldwide electricity used in data centers", Environmental Research Letters **3**, Jul-Sep 2008

Google

Net IT Carbon Footprint: Strongly Negative!

The carbon footprint of information pays for itself...and then some



*source: GeSI/The Climate Group: SMART 2020: Enabling the low carbon economy in the information age

Efficiency of warehouse-scale computing: carbon

0.2g Answering one **Google query**

20g Using a **Laptop** for one hour

75g Using a **PC & monitor** for one hour

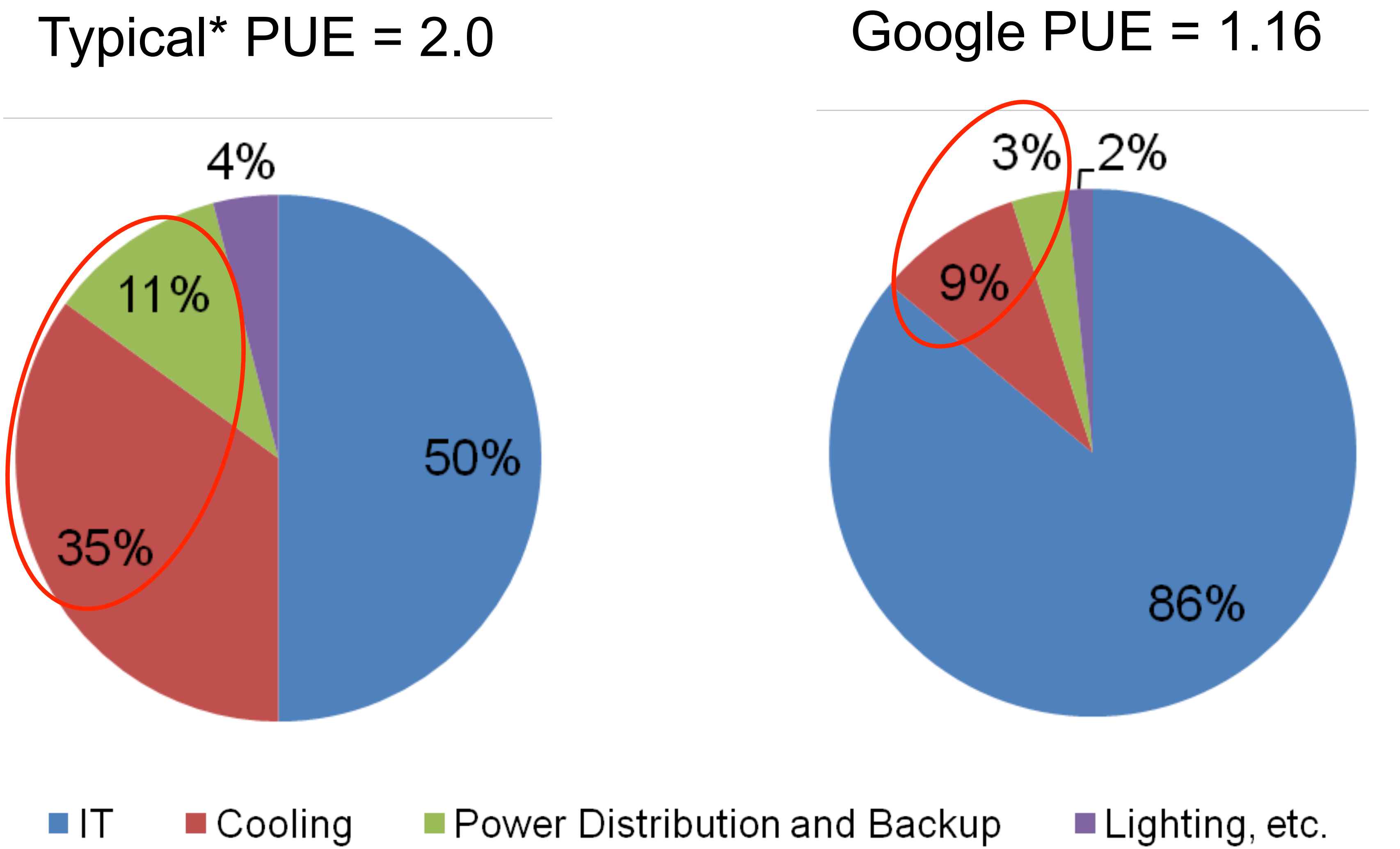
173g One weekday **newspaper** (physical copy)

209g Producing a single glass of **orange juice**

280g Washing one load of **laundry** in an efficient machine

532g One **beer**

Dramatic reduction of overheads



*Reference: Silicon Valley Leadership Group, Data Center Energy Forecast, Final Report July, 2008

Google E Data Center energy-weighted average PUE results from Q2-Q1'09 (to 3/15/09)

Google

Fault Recovery

- 99.9% uptime = 9 hours down/year
- A 10,000 server warehouse can expect
  - 0.25 cooling/power failure (all down; day)
  - 1 PDU failure (500 down; 6 hours)
  - 20 rack failures (40 down; 1 hour)
  - 3 router failures (1 hour)
  - 1000 server failure
  - 1000s disk failures
  - etc., etc., etc.

Stuff happens ...

Stuff happens ...

- Power failures

Stuff happens ...

- Power failures
- Cosmic rays

Stuff happens ...

- Power failures
- Cosmic rays
- Software bugs

Stuff happens ...

- Power failures
- Cosmic rays
- Software bugs
- Thieves

Stuff happens ...

- Power failures
- Cosmic rays
- Software bugs
- Thieves
- Drunken hunters

Stuff happens ...

- Power failures
- Cosmic rays
- Software bugs
- Thieves
- Drunken hunters
- Sharks

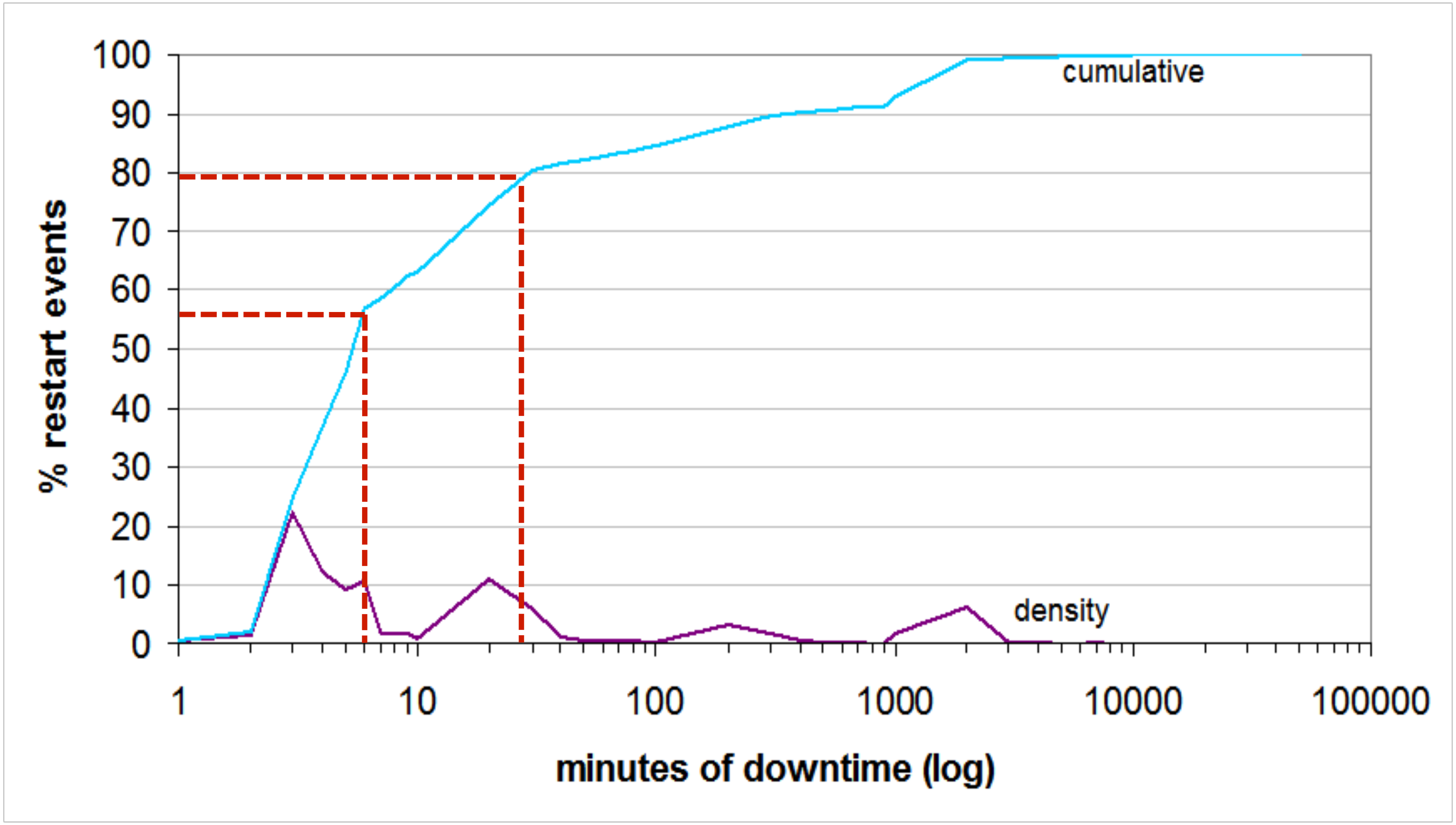
Stuff happens ...

- Power failures
- Cosmic rays
- Software bugs
- Thieves
- Drunken hunters
- Sharks
- Blasphemy

Stuff happens ...

- Power failures
- Cosmic rays
- Software bugs
- Thieves
- Drunken hunters
- Sharks
- Blasphemy
- ...

Understanding downtime behavior matters



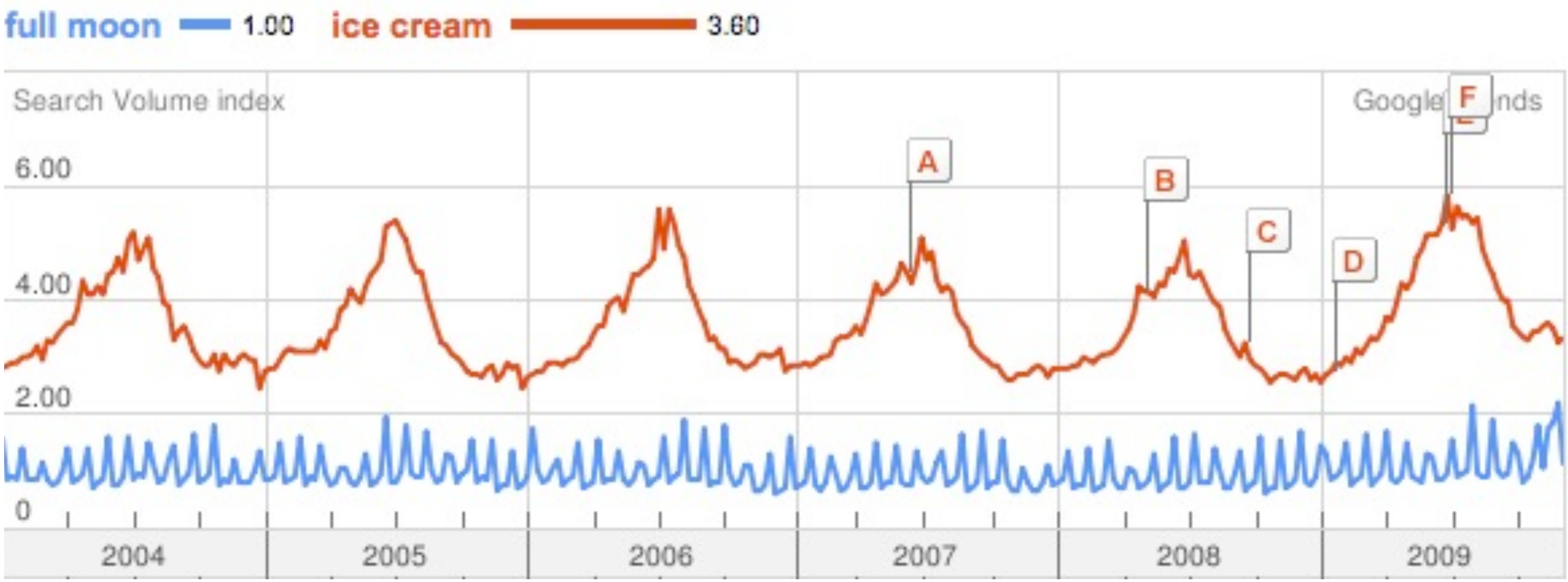
Google

Planning for Recovery

- Replication
- Sharding
- Checkpoints
- Monitors / Heartbeats
- If possible:
  - Loose consistency
  - Approximate answers
  - Incomplete answers

George Box (1919-)

Essentially,
all models are wrong,
but some are useful.


full moon 1.00
ice cream 3.60
Search Volume index
Google Trends
6.00
4.00
2.00
0
A
B
C
D
F
2004
2005
2006
2007
2008
2009

Explore flu trends - United States

We've found that certain search terms are good indicators of flu activity. Google Flu Trends uses aggregated Google search data to estimate flu activity. Learn more »

National

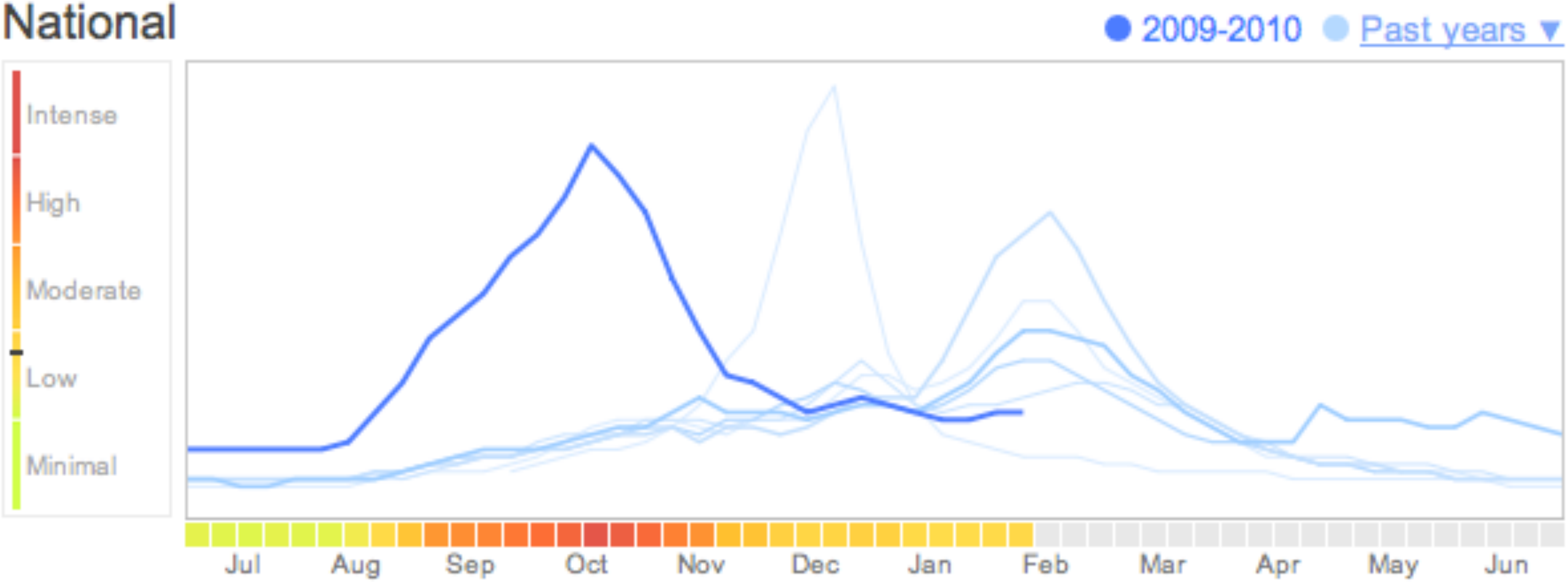


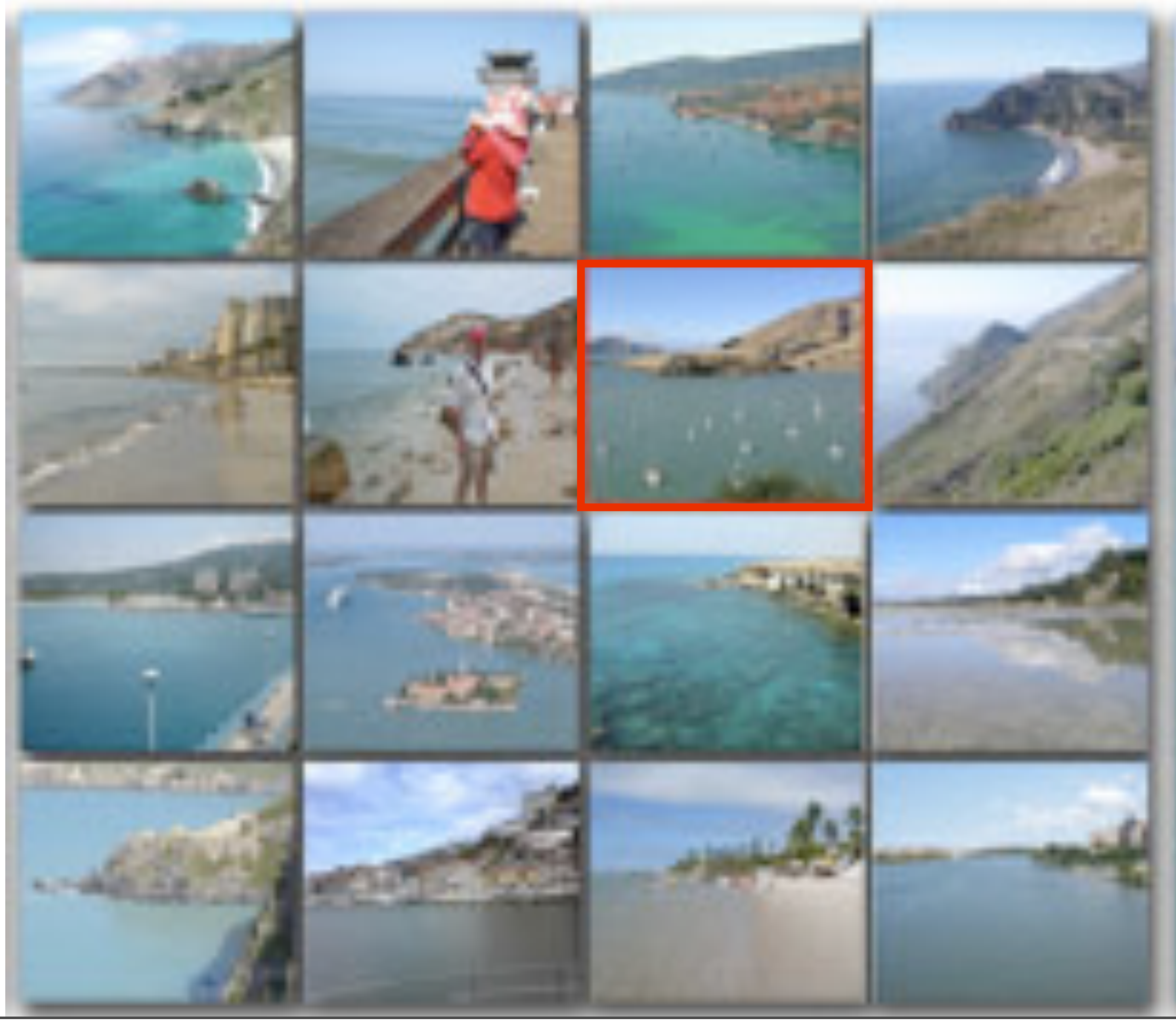
Image Models

Cave painting, Lascaux, France, 15,000 to 10,000 B.C.

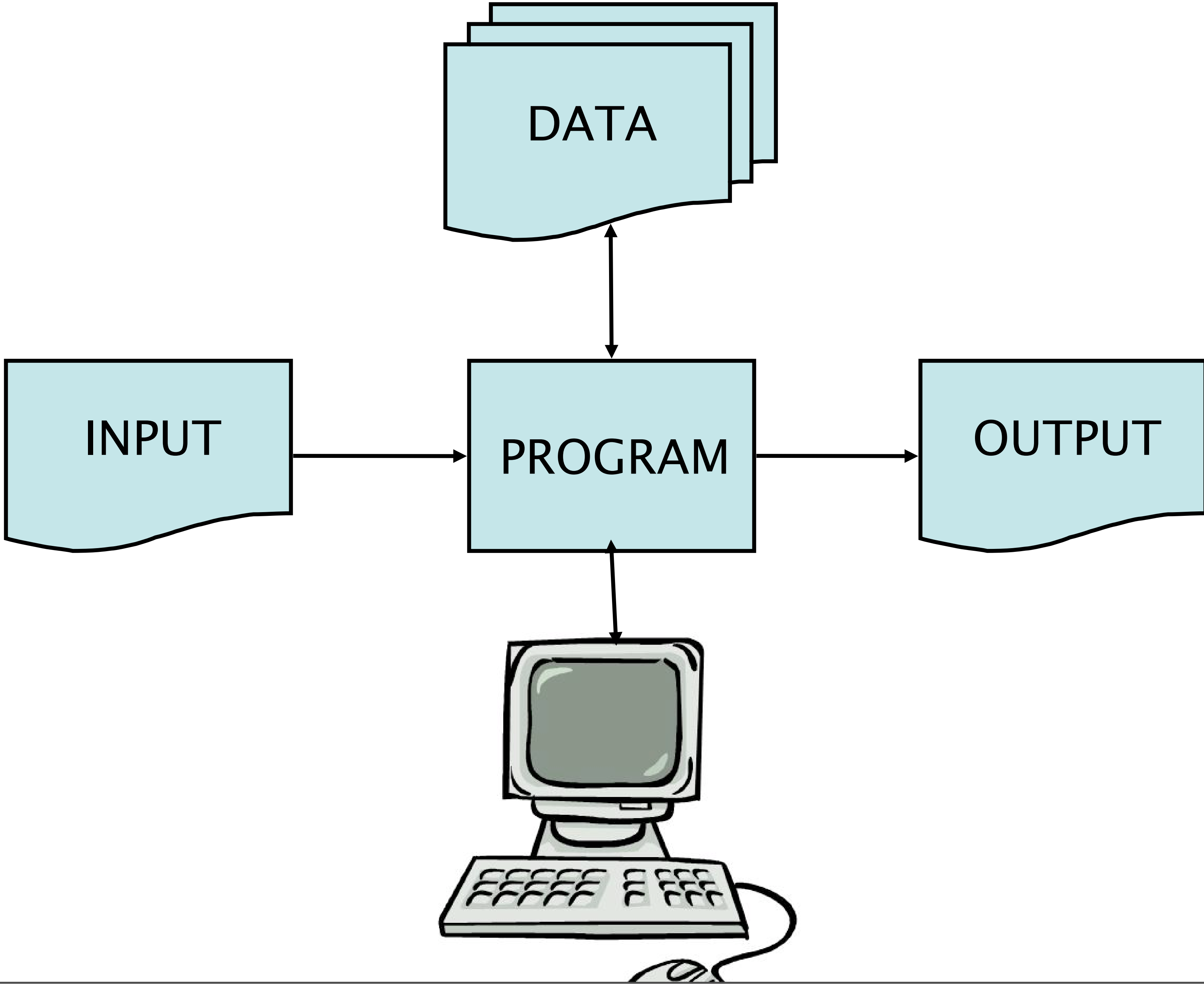
James Hays, Alexei Efros, CMU: Scene Completion

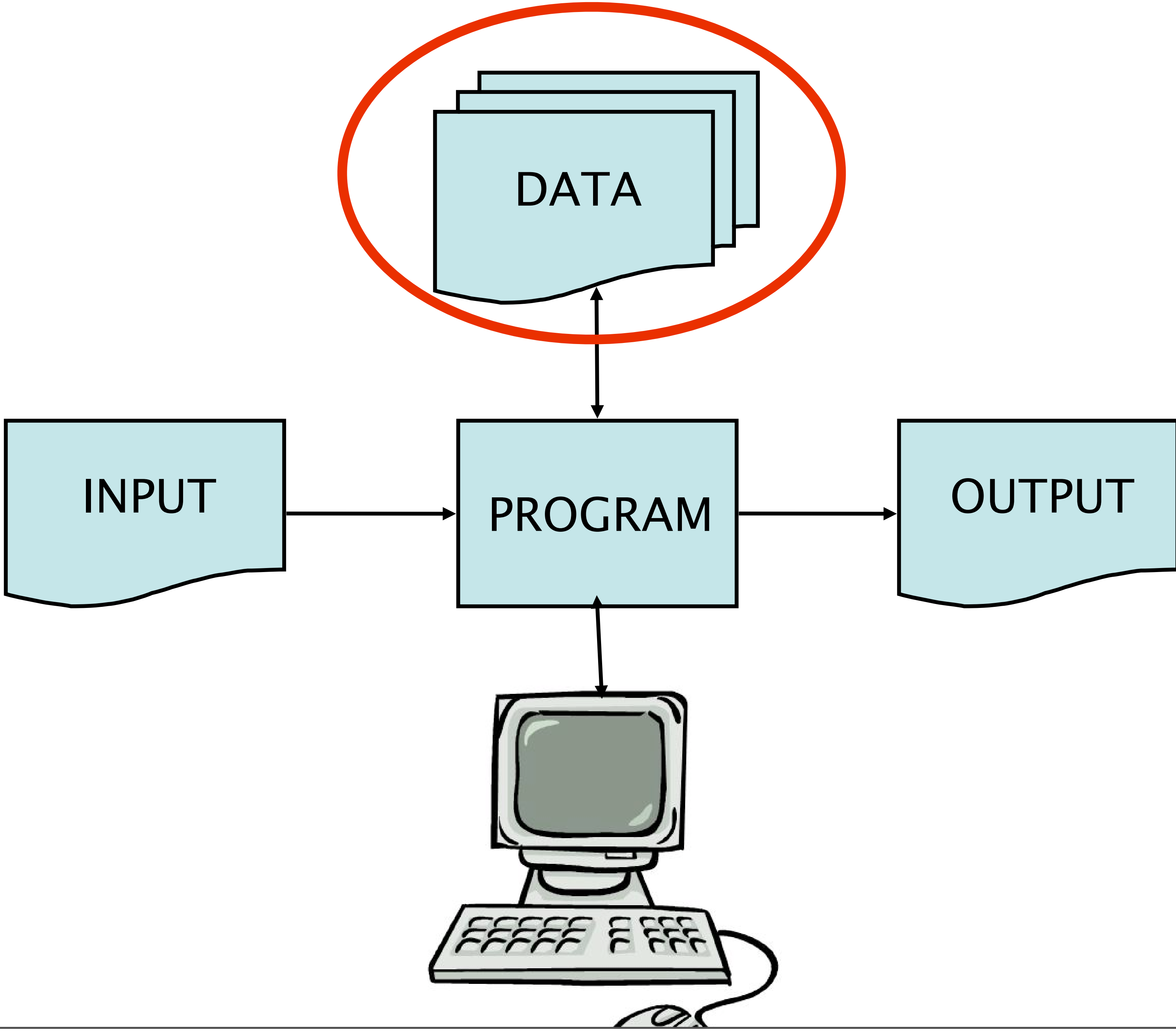
James Hays, Alexei Efros, CMU: Scene Completion

James Hays, Alexei Efros CMU: Scene Completion

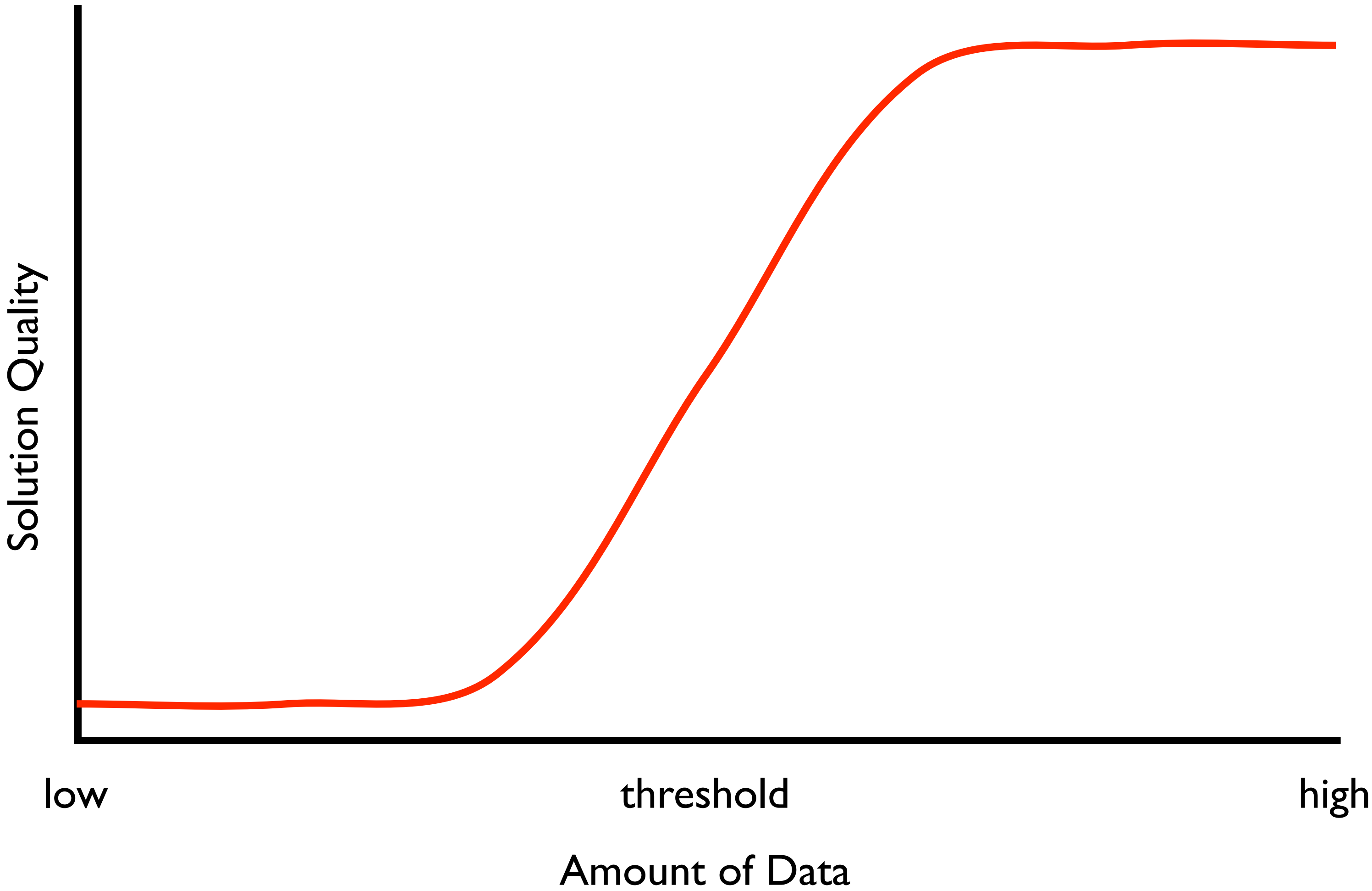


James Hays, Alexei Efros CMU: Scene Completion

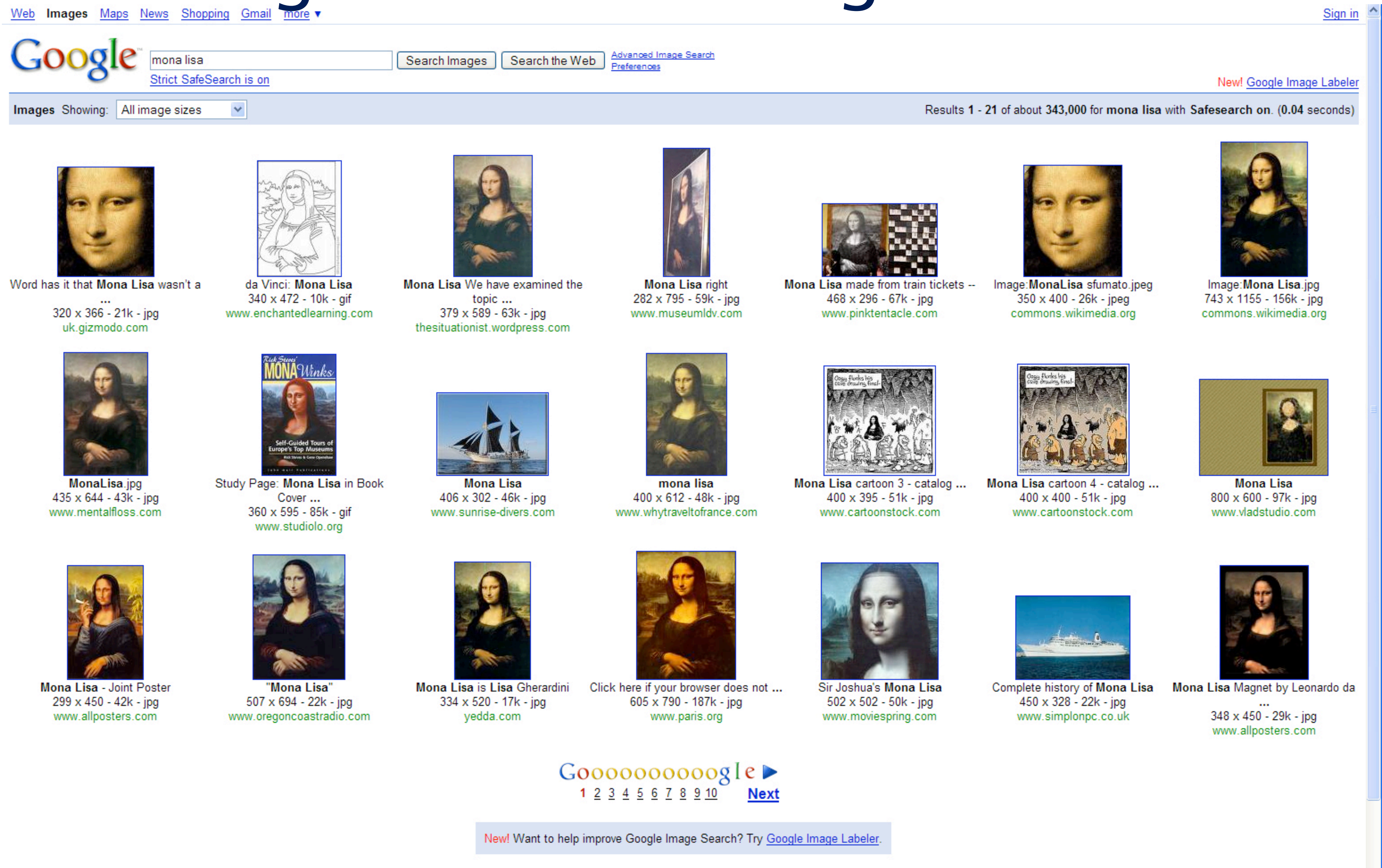

DATA
INPUT
PROGRAM
OUTPUT


DATA
INPUT
PROGRAM
OUTPUT

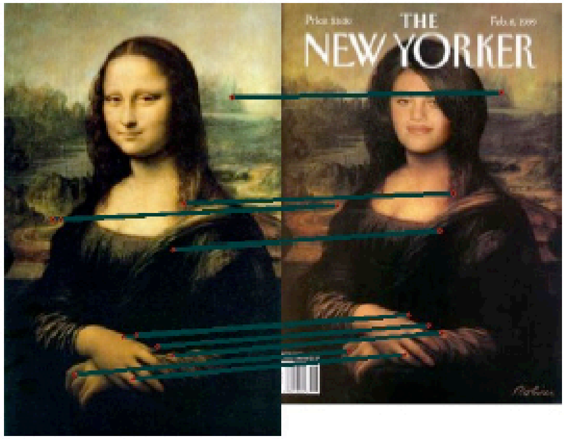
Data Threshold



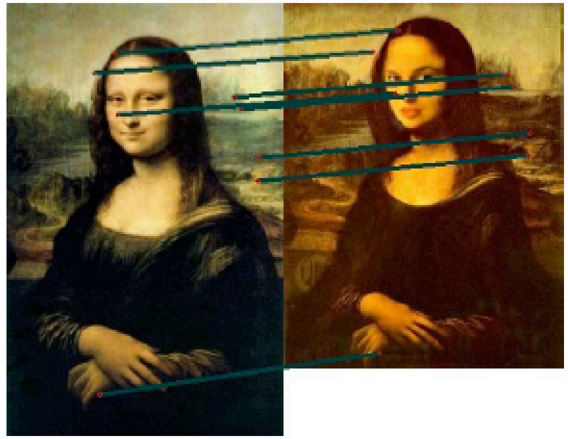
Jing, Baluja, Rowley, Google: Finding Canonical Images



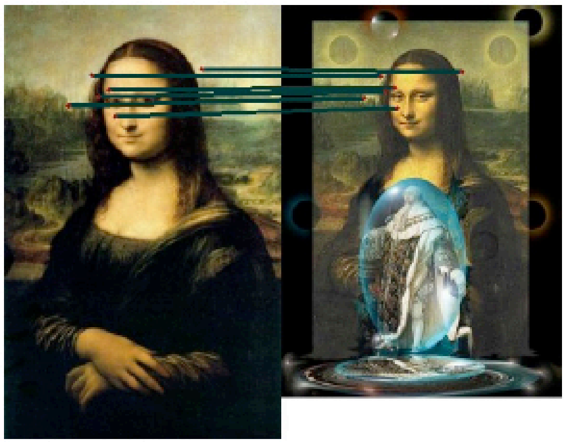
Compare low-level features



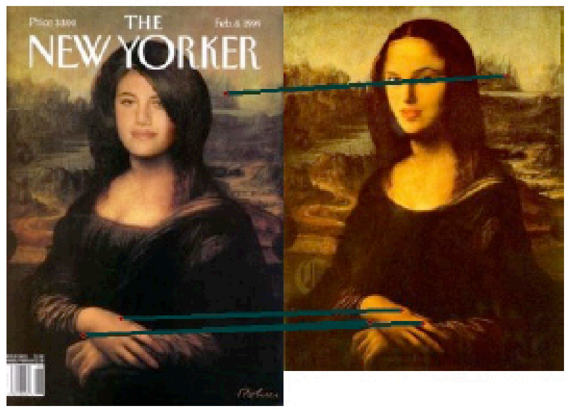
(a) A v.s. B



(b) A v.s. C

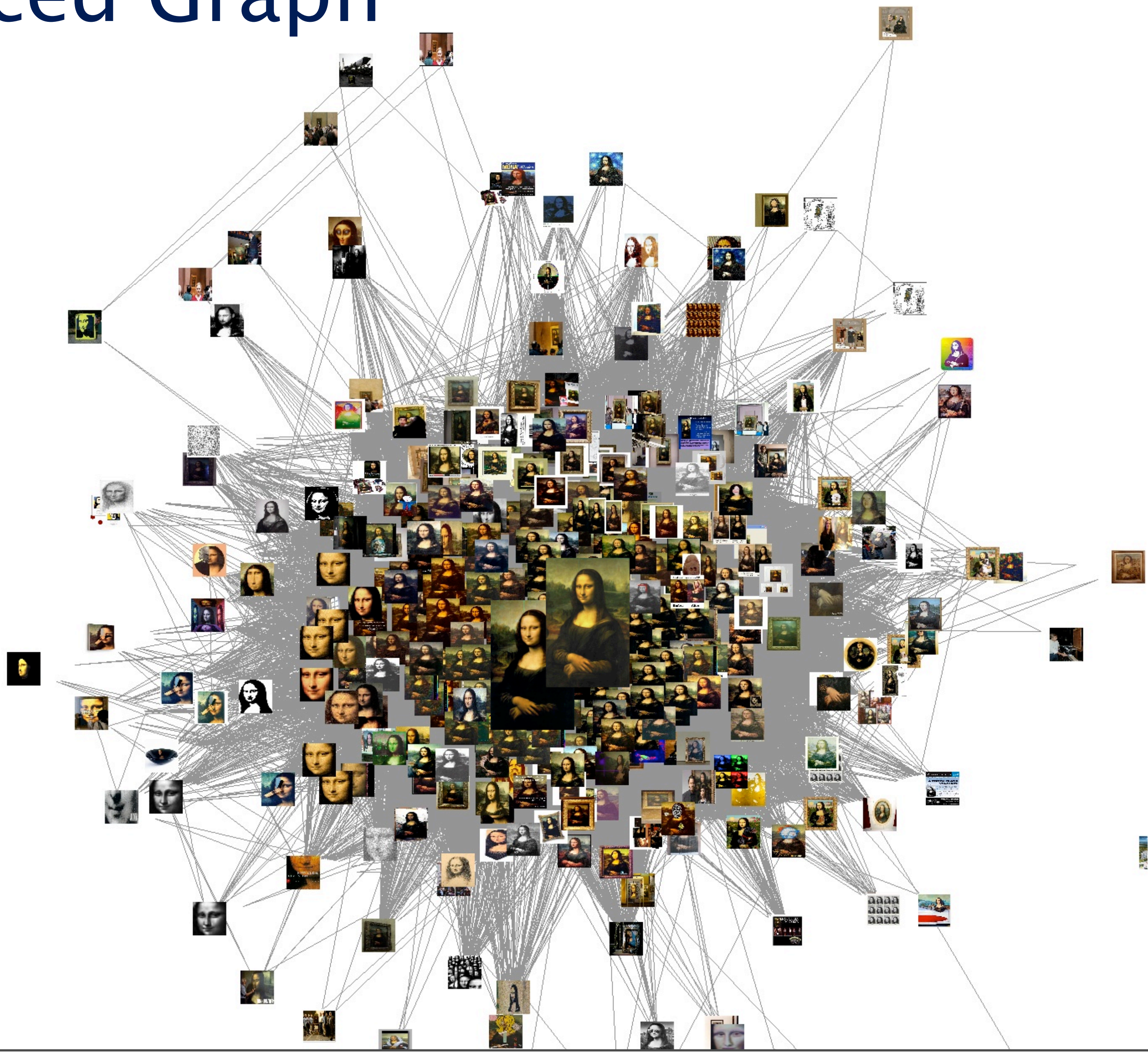


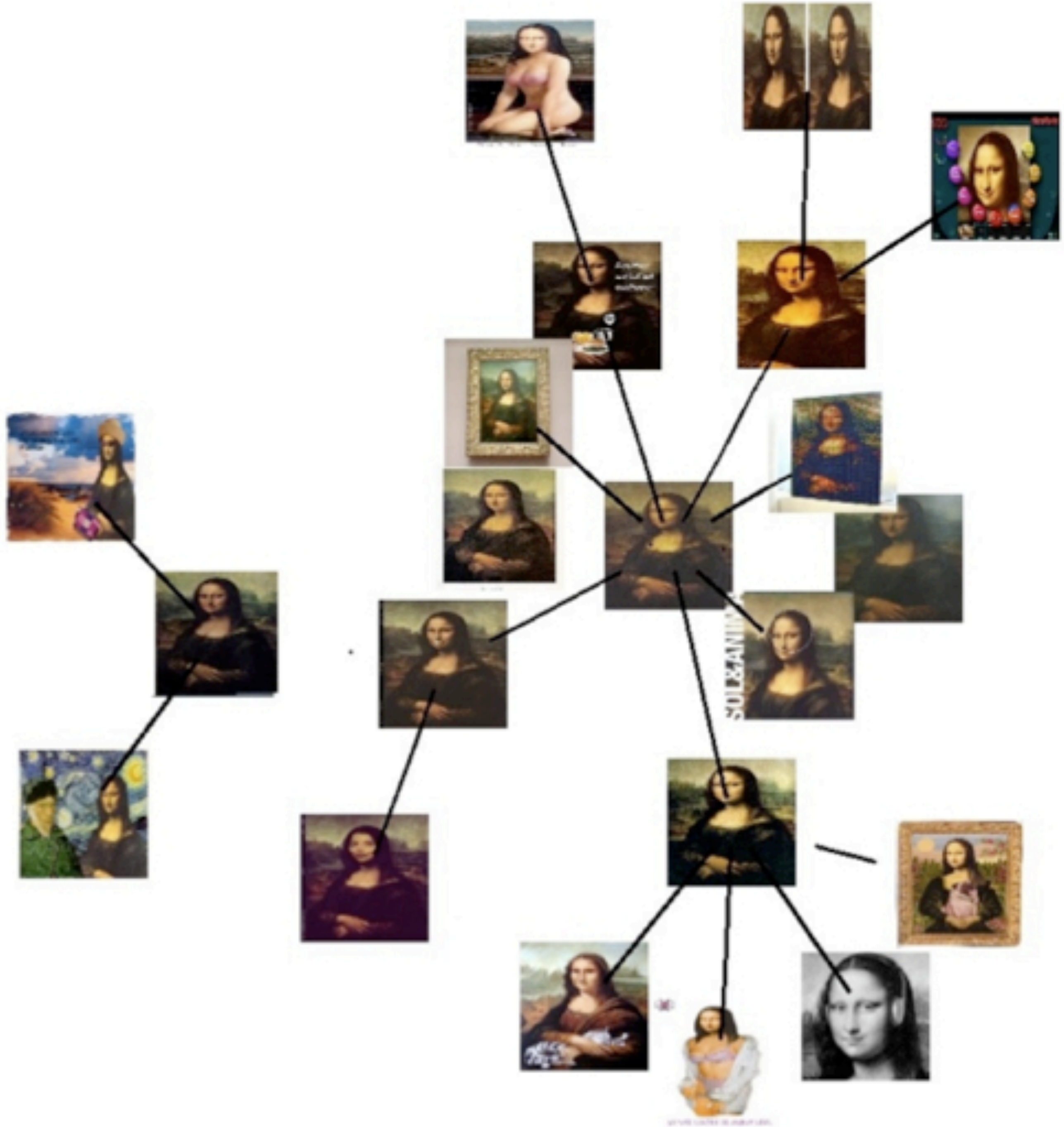
(c) A v.s. D



(d) B v.s. C

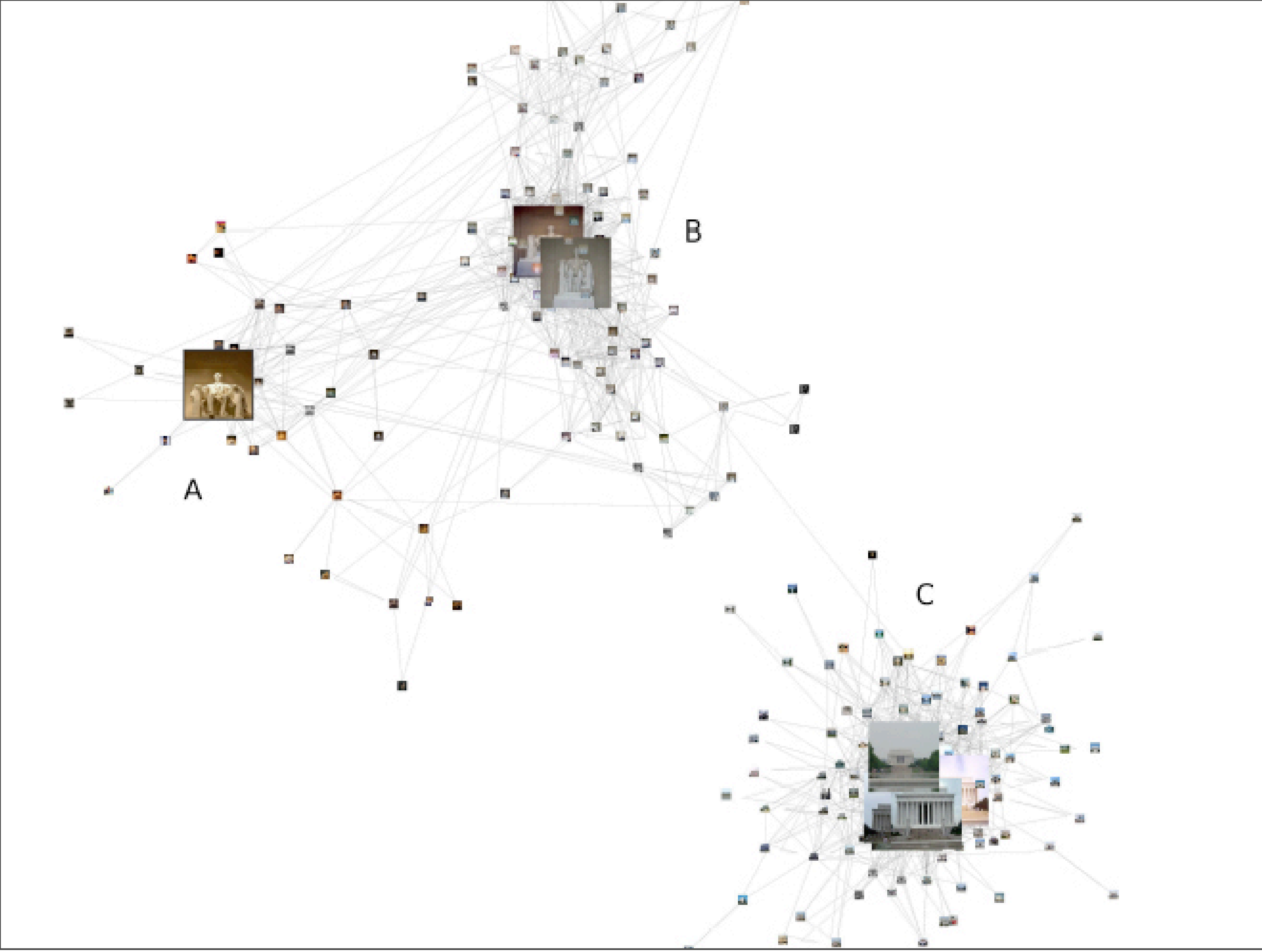
Induced Graph






STARBUCKS COFFEE
ORACLE

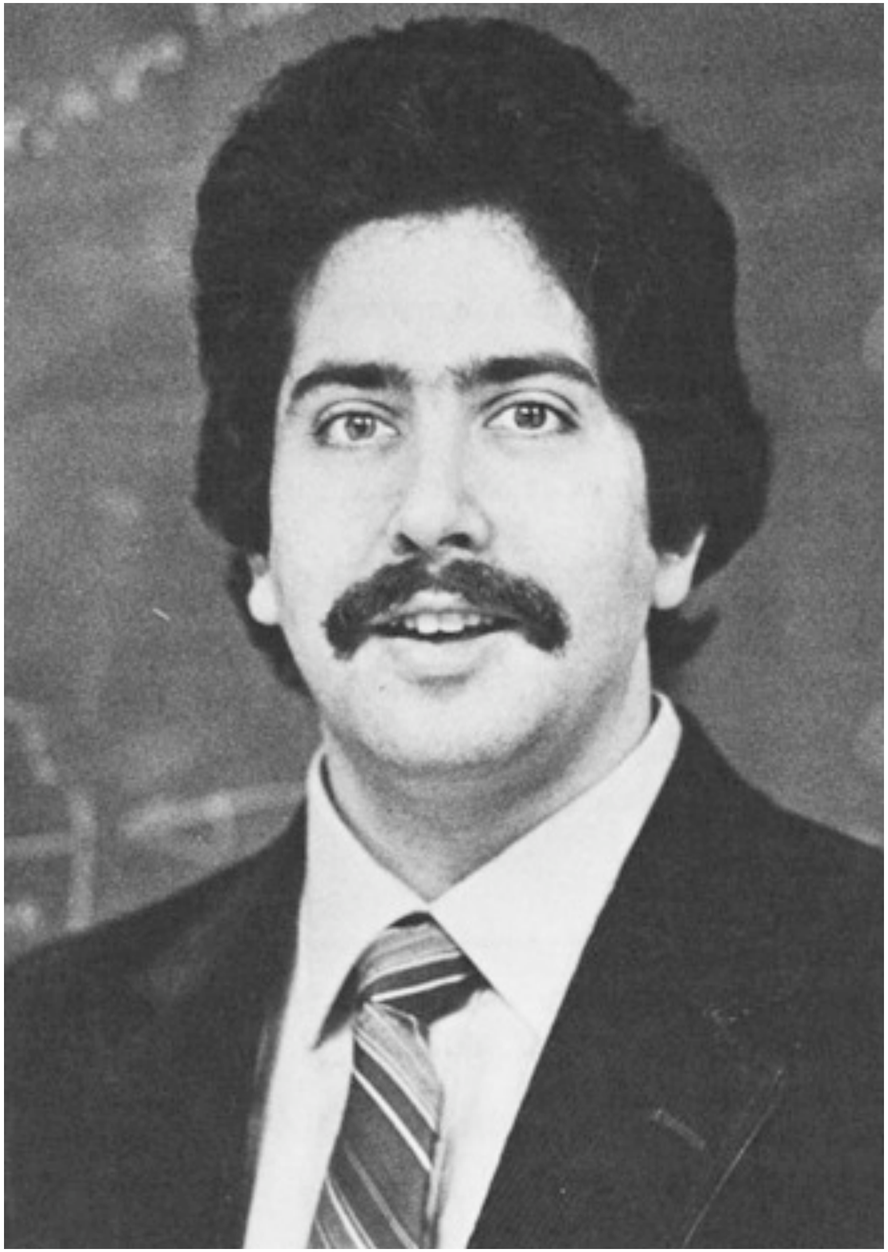
SIFT Features


B
A
C

Text Models

Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,

Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,
Lorum ipsum dolor sit amet, consectetur elit, Lorum ipsum dolor sit amet, consectetur elit,



Doug Lenat (1950-)

Ed Feigenbaum (1936-)

Lenat and Feigenbaum, 1991:

Each of us has a vast storehouse of general knowledge, though we **rarely** talk about any of it explicitly to one another; we just assume that other people already know these things. If they are included in a conversation, or an article, they confuse more than they clarify. Some examples are:

- **Water flows downhill**
- **Living things get diseases**
- …”

Lenat and Feigenbaum, 1991:

Each of us has a vast storehouse of general knowledge, though we **rarely** talk about any of it explicitly to one another; we just assume that other people already know these things. If they are included in a conversation, or an article, they confuse more than they clarify. Some examples are:

- **Water flows downhill**
- **Living things get diseases**
- …"

$10,000 per page

Lenat and Feigenbaum, 1991:

Each of us has a vast storehouse of general knowledge, though we **rarely** talk about any of it explicitly to one another; we just assume that other people already know these things. If they are included in a conversation, or an article, they confuse more than they clarify. Some examples are:

- **Water flows downhill**
- **Living things get diseases**
- …"

$10,000
per page

Lenat and Feigenbaum, 1991:

Ea s a vast storehouse of general knowledge, though w about any of it explicitly to one another; we just a ther people already know these things. If they are in onversation, or an article, they confuse more tha rify. Some examples are:

- **Water flows downhill**
- **Living things get diseases**
- …"

rarely = 0.0001% of pages

$10,000 per page

1 - 10 of about 16,600 for "water flows downhill". (0.29 seconds)

Water Flows Downhill:: Lesson Plan, Activity, or Teaching Idea ...
Children will experiment with different containers to see if water flows up or down.
www.atozteacherstuff.com/pages/515.shtml - 32k - Cached - Similar pages

Word Sense Disambiguation

bank[1] |ba NG k|

noun

1 the land alongside or sloping down to a river or lake : *willows lined the riverbank.*

bank[2] |bøŋk| |baŋk|

noun

a financial establishment that invests money deposited by customers, pays it out when required, makes loans at interest, and exchanges currency : *I paid the money straight into my bank.*

More Data vs. Better Algorithms

Banko & Brill, 2001

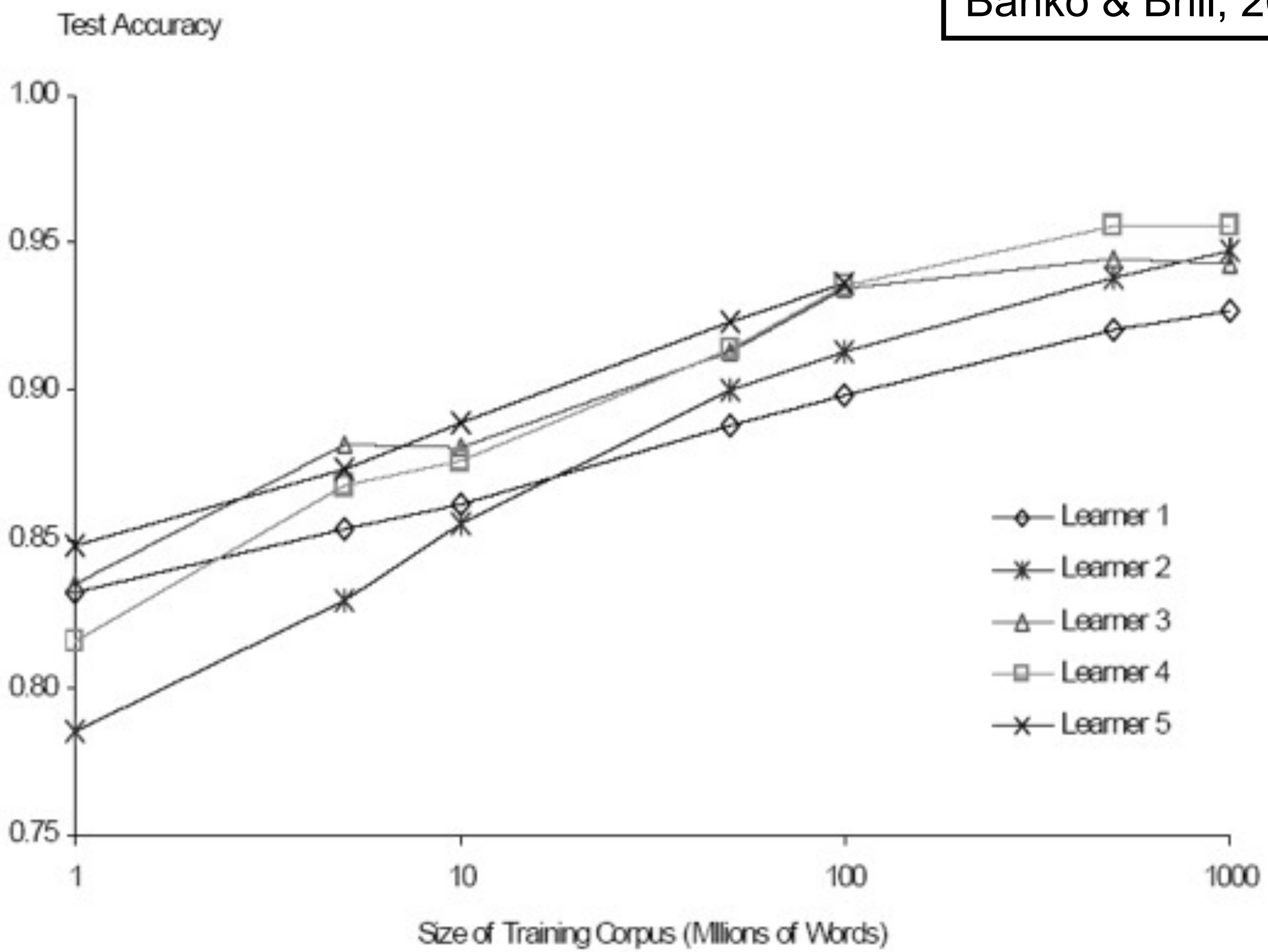


Figure 2. Learning Curves for Confusable Disambiguation

More Data vs. Better Algorithms

Banko & Brill, 2001

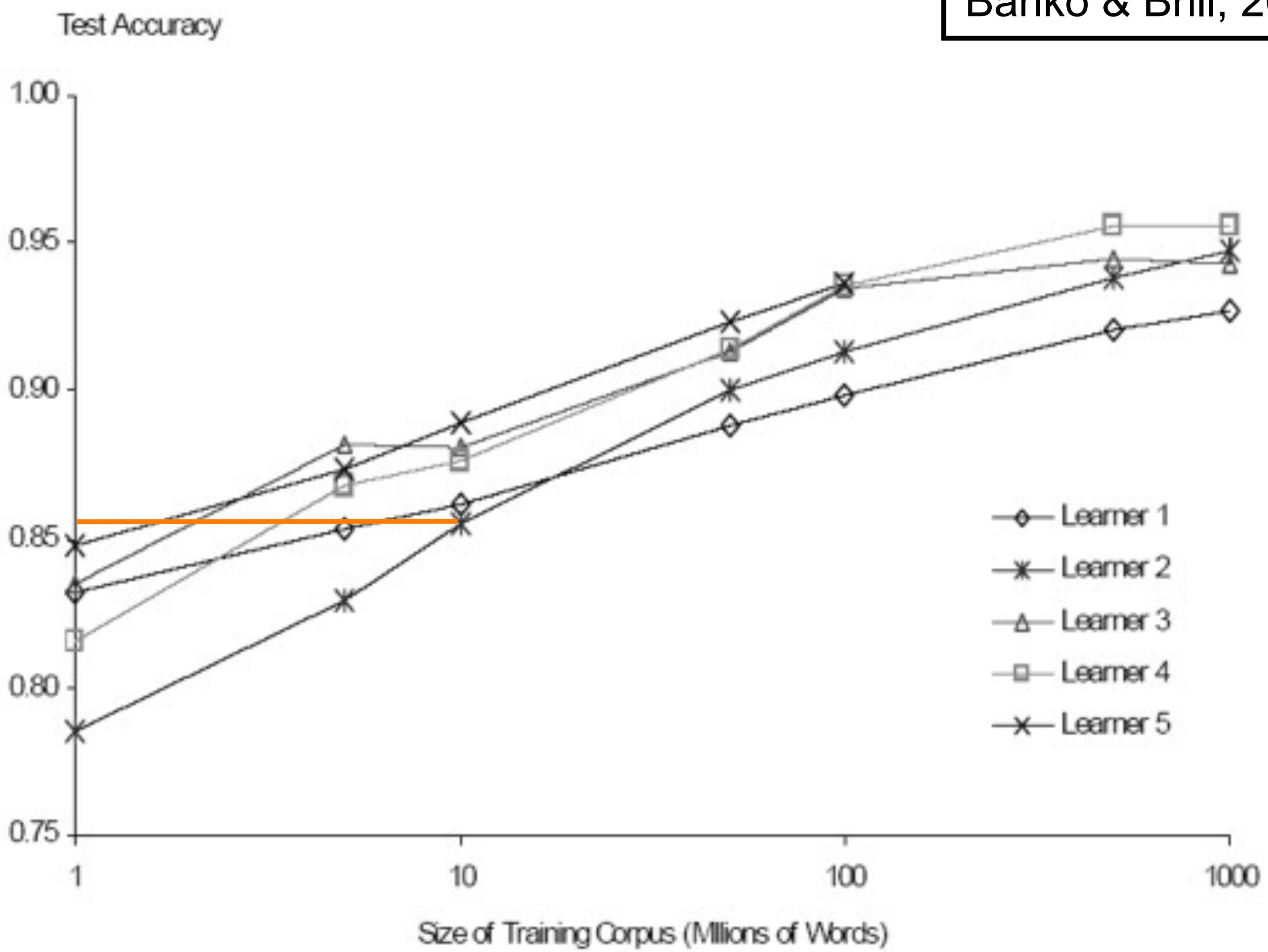


Figure 2. Learning Curves for Confusable Disambiguation

More Data vs. Better Algorithms

Banko & Brill, 2001

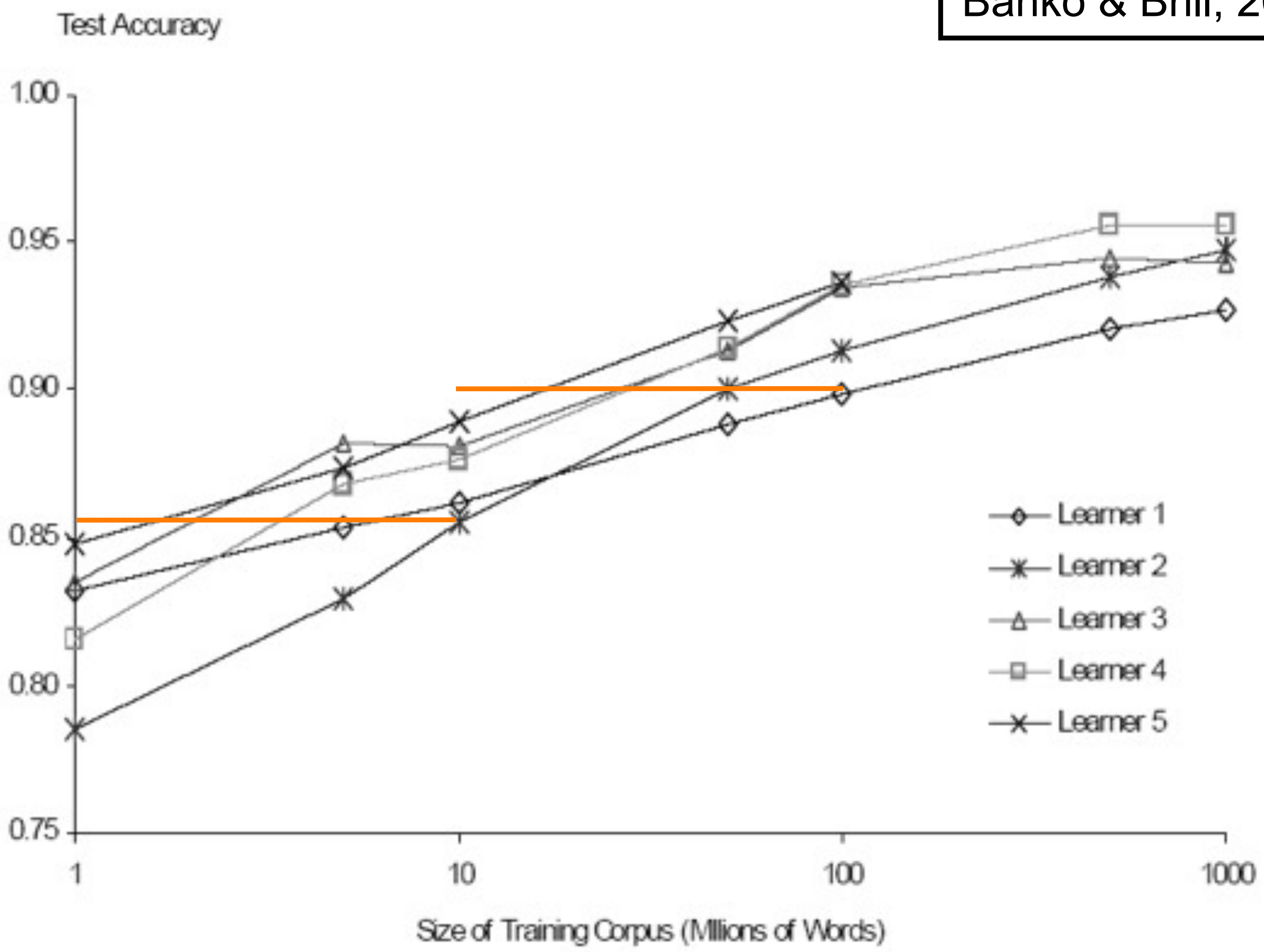


Figure 2. Learning Curves for Confusable Disambiguation

More Data vs. Better Algorithms

Banko & Brill, 2001

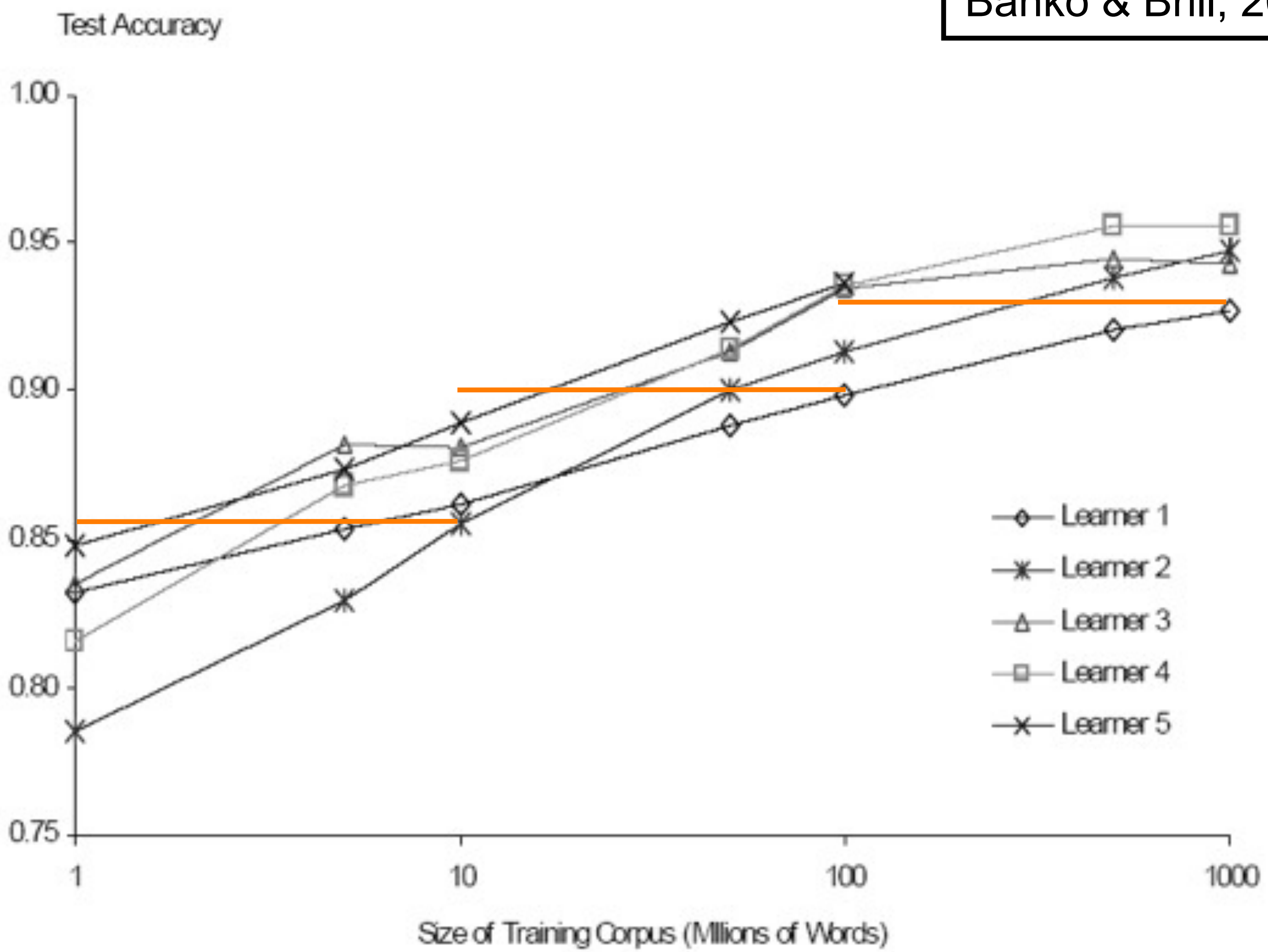


Figure 2. Learning Curves for Confusable Disambiguation

Spelling

Mehran Sahami

Spelling

- Dictionary Based:

Mehran Sahami

Spelling

- Dictionary Based:

Tehran Salami

Spelling

- Dictionary Based:

Tehran Salami

- Corpus Based:

Mehran Sahami: ok
Mehron Sahami
Did you mean: **Mehran** Sahami

Spelling

Spelling

$$\begin{aligned} best &= \text{argmax}_c \; \mathrm{P}(c \mid w) \\ &= \text{argmax}_c \; \mathrm{P}(w \mid c)\, \mathrm{P}(c) \end{aligned}$$

Spelling

$$best = \mathrm{argmax}_c \; P(c \mid w)$$
$$= \mathrm{argmax}_c \; P(w \mid c) \; P(c)$$

Spelling

$$best = \mathrm{argmax}_c\ \mathrm{P}(c \mid w)$$
$$= \mathrm{argmax}_c\ \mathrm{P}(w \mid c)\ \mathrm{P}(c)$$

$\mathrm{P}(c)$ ~ estimated from word counts

Spelling

$$best = \text{argmax}_c \, P(c \mid w)$$
$$= \text{argmax}_c \, P(w \mid c) \, P(c)$$

$P(c)$ ~ estimated from word counts

$P(w \mid c)$ ~ proportional to edit distance

```
import string, collections
def train(filename):
    P = collections.defaultdict(lambda: 1)
    for line in file(filename):
        word, count = line.split()
        P[word] = int(count)
    return P
P = train('en100k.txt')
def edits1(word):
    n = len(word)
    return set([word[0:i]+word[i+1:] for i in range(n)] + # deletion
               [word[0:i]+word[i+1]+word[i]+word[i+2:] for i in range(n-1)] + # transposition
               [word[0:i]+c+word[i+1:] for i in range(n) for c in string.lowercase] + # alteration
               [word[0:i]+c+word[i:] for i in range(n+1) for c in string.lowercase]) # insertion
def known_edits2(word):
    return set(e2 for e1 in edits1(word) for e2 in edits1(e1) if e2 in P)
def known(words):
    return set(w for w in words if w in P)
def correct(word):
    candidates = known([word]) or known(edits1(word)) or known_edits2(word) or [word]
    return argmax(candidates, P)
```

http://www.htdig.org/ files/ htdig-3.2.0b5.tar.bz2/ htdig-3.2.0b5/ htfuzzy/

Files | Outline New!

- ..
- Accents.cc
- Accents.h
- Endings.cc
- Endings.h
- EndingsDB.cc
- Exact.cc
- Exact.h
- Fuzzy.cc
- Fuzzy.h
- Makefile.am
- Makefile.in
- Makefile.win32
- **Metaphone.cc**
- Metaphone.h
- Prefix.cc
- Prefix.h
- Regexp.cc
- Regexp.h
- Soundex.cc
- Soundex.h
- Speling.cc
- Speling.h
- Substring.cc
- Substring.h
- SuffixEntry.cc
- SuffixEntry.h
- Synonym.cc
- Synonym.h
- htfuzzy.cc

Metaphone.cc

```
    for (; *n && key.length() < MAXPHONEMELEN; n++)
    {
        /* Drop duplicates except for CC */
        if (*(n - 1) == *n && *n != 'C')
          continue;
        /* Check for F J L M N R or first letter vowel */
        if (same(*n) || *(n - 1) == '\0' && vowel(*n))
          key << *n;
        else
        {
            switch (*n)
            {
            case 'B':
                /*
                 * B unless in -MB
                 */
                if (*(n + 1) || *(n - 1) != 'M')
                    key << *n;
                break;
            case 'C':
                /*
                 * X if in -CIA-, -CH- else S if in
                 * -CI-, -CE-, -CY- else dropped if
                 * in -SCI-, -SCE-, -SCY- else K
                 */
                if (*(n - 1) != 'S' || !frontv(*(n + 1)))
                {
                    if (*(n + 1) == 'I' && *(n + 2) == 'A')
                        key << 'X';
                    else if (frontv(*(n + 1)))
                        key << 'S';
                    else if (*(n + 1) == 'H')
                        key << (((*(n - 1) == '\0' && !vowel(*(n + 2)))
                                 || *(n - 1) == 'S')
                                ? 'K' : 'X');
                    else
                        key << 'K';
                }
```

Google LDC N-Gram Corpus

Number of tokens:	1,024,908,267,229
Number of sentences:	95,119,665,584
Number of unigrams:	13,588,391
Number of bigrams:	314,843,401
Number of trigrams:	977,069,902
Number of fourgrams:	1,313,818,354
Number of fivegrams:	1,176,470,663

Google LDC N-Gram Corpus

serve as the independent 794
serve as the index 223
serve as the indication 72
serve as the indicator 120
serve as the indicators 45
serve as the indispensable 111
serve as the indispensible 40
serve as the individual 234
serve as the industrial 52
serve as the industry 607

Wordnet Dictionary

Verb

- S: (v) **serve**, function (serve a purpose, role, or function) *"The tree stump serves as a table"; "The femal serve very well"; "His freedom served him well"; "The table functions as a desk"*
- S: (v) **serve** (do duty or hold offices; serve in a specific function) *"He served as head of the department f*
- S: (v) **serve** (contribute or conduce to) *"The scandal served to increase his popularity"*
- S: (v) service, **serve** (be used by; as of a utility) *"The sewage plant served the neighboring communities"*
- S: (v) **serve**, help (help to some food; help with food or drink) *"I served him three times, and after that h*
- S: (v) **serve**, serve up, dish out, dish up, dish (provide (usually but not necessarily food)) *"We serve mea P.M."; "The entertainers served up a lively show"*
- S: (v) **serve** (devote (part of) one's life or efforts to, as of countries, institutions, or ideas) *"She served the country"*
- S: (v) **serve**, serve well (promote, benefit, or be useful or beneficial to) *"Art serves commerce"; "Their i President's wisdom has served the country well"*
- S: (v) **serve**, do (spend time in prison or in a labor camp) *"He did six years for embezzlement"*
- S: (v) **serve**, attend to, wait on, attend, assist (work for or be a servant to) *"May I serve you?"; "She atte our table, please?"; "Is a salesperson assisting you?"; "The minister served the King for many years"*
- S: (v) **serve**, process, swear out (deliver a warrant or summons to someone) *"He was processed by the s*
- S: (v) suffice, do, answer, **serve** (be sufficient; be adequate, either in quality or quantity) *"A few words w $100 do?"; "A 'B' grade doesn't suffice to get me into medical school"; "Nothing else will serve"*
- S: (v) **serve** (do military service) *"She served in Vietnam"; "My sons never served, because they are sho*
- S: (v) **serve**, service (mate with) *"male animals serve the females for breeding purposes"*
- S: (v) **serve** (put the ball into play) *"It was Agassi's turn to serve"*

WordNet home page