CSCI 572
Information Retrieval and Web Search Engines

Prof. Ellis Horowitz

General Requirements

- All lectures will be online and available via DEN
  - https://courses.uscden.net/
  - Email me if you wish to attend my recording of the lecture
- Both exams are open book and available online, there is no final exam
  - Exam 1, Sept. 29th
  - Exam 2, Dec. 1st
- TA/grader interactions and support will be offered via Piazza
- There are five homework assignments with the last one a comprehensive project

Copyright Ellis Horowitz, 2011 - 2022

The Class Goals

- **This course will give you**
  - **a broad understanding of how search engines work, and**
  - **will equip you to participate in designing and developing search engine capabilities**
  - **while covering the basic elements of information retrieval**
- **It discusses the fundamental building blocks of web search engines including:**
  - **crawling the web**
  - **indexing the data for fast retrieval**
  - **query processing including special ranking algorithms e.g. PageRank and HITS**
  - **a detailed analysis of the Google Search Engine including: Google File System and Map/Reduce**

Copyright Ellis Horowitz, 2011 - 2022

Special Acknowledgments

- The following courses cover similar material and I wish to acknowledge some of the instructors for making their notes available:
- Stanford University
  - CS 276, Information Retrieval and Web Search, Spring 2019, Manning and Nayak
  - http://web.stanford.edu/class/cs276/
- Univ. of Texas, Austin
  - CS 371R, Information Retrieval and Web Search, Ray Mooney, Fall 2019
- Univ. of Mass.
  - CMPSCI 646, Information Retrieval, Spring 2014, James Allan, http://ciir.cs.umass.edu/~allan/cs646/
- Cornell University
  - https://www.cs.cornell.edu/courses/cs4300/2013fa/lectures.htm
- Johns Hopkins
  - 605.766 Information Retrieval, Fall 2019, Prof. Paul McNamee, pmcnamee.net/ir.html
- Purdue University
  - CS 54701, Information Retrieval Fall, 2019 http://cs.purdue.edu/homes/clifton/cs54701/
- Penn State
  - IST 441, Prof. C. Lee Giles, Spring, 2019, http://clgiles.ist.psu.edu/IST441/
- Also, the following textbooks provided a variety of material
  - *Mining Massive Data* Sets, A. Rajaraman and J. Ullman, 2011
  - *Introduction to Information Retrieval* by C. Manning, P. Raghavan, H. Schutze, 2009
  - *Mining the Web*, S. Chakrabarti, 2003

Copyright Ellis Horowitz, 2011 - 2022

Course Website

http://csci572.com/

Home Page

Course Lectures

CSCI 572 Home Page

Not Secure | csci572.com

CSCI 572 Home P... | Piazza Spring2022 | CSCI572_Spring2... | DEN D2L Page | USC Schedule of... | Computer Science... | Other Bookmarks

- Home Page
- Recent Search Engine Articles
- Schedule of Lectures
- Assignments
- Special Resources
- Course Grading
- Course Materials
- Class News Group Piazza
- Academic Integrity Policy
- How to Submit Your Homework

Schedule of Lectures

Please observe that all of these notes are copyrighted and are for the exclusive use of students enrolled in CS572. Please do not make the login&password available to others, and please do not distribute copies to others.

Date	Topic	Course Notes	Readings from Introduction to Information Retrieval, Manning, Raghavan, Schutze, 581 page book, pdf All GOTO links below refer to chapters/sections in the above book.	Video Lectures, some taken from • Information Retrieval Course Matching Our Textbook • 8th International Summer School on Information Retrieval NOTE: Videos with a double star (**) are required for the exam; You are only responsible for videos with **
Week 1 Aug 23	A: Introduction to the Course B: Search Engine Basics	PPT, PDF PPT, PDF	How Google Fights Disinformation	
Week 1 Aug 25	Characterizing the Web **Discussion of Homework #1**	PPT, PDF		• How Google Founders Changed The World (start at 4:16 and watch till bored) Other useful videos • WDM1:What is Data Mining (8 min) • WDM2:Structured Data and IR (17 min)

- I will normally lecture for 1 hour;
- We will then watch one or more relevant videos
- All slides are available in PPT and PDF
- Login/password for the class notes are: csci572 notes

Course Grading

- **50%, 5 class programming assignments each worth 10%;**
- **50% two exams;**
- **Re-grading opportunities only for those scoring 5 *or more points below the average***

LIMITED Re-Grading Opportunities of Exams

Unfortunately, due to the size of the class there will be **NO** opportunity for all to see your exam and have it re-graded. ***Please do not take this class*** if you do not agree with this policy

Copyright Ellis Horowitz, 2011 - 2022

Special Resources

Videos,
Data sets,
Algorithms,
Papers

CSCI 572 Home Page

Not Secure | csci572.com

Apps | CSCI 572 Home P... | CSCI 571 - Home | Computer Science... | ITS - Software | Masters Student... | University of Sout... | Other Bookmarks

- Home Page
- Recent Search Engine Articles
- Schedule of Lectures
- Assignments
- Special Resources
- Course Grading
- Course Materials
- Class News Group Piazza
- Your Grades
- Academic Integrity Policy

Selected Videos | Selected DataSets | Selected Algorithms | Selected Papers

Selected Videos

- Problems with Google Search, TED talk, 2011(9 min)
- Google Developers Website, Numerous Useful Videos
- Google's Tech Talk Channel on YouTube (lots of general subjects)
- Challenges in Building Large-Scale Information Retrieval Systems by Jeff Dean, Google Fellow, Feb. 2009, (1hr, 5min)
- Machine Learning in Information Retrieval by Thomas Hoffman, a 4 hour tutorial, Sept. 2004
- Some Key Challenges in Web Crawlers and Content-Based Search Engines, July 2006, (27min)
- Predicting the Present with Search Engine Data, Hal Varian, Aug. 2013, (51min)
- Experiences with the Nutch search engine by Doug Cutting, July, 2006, (1hr, 11min)
- Building blocks for semantic search engines by Soumen Chakrabarti, July 2006, (1hr 4min)
- Inside Google's Search Office, (1 hr 27 min)
- The Structured Search Engine by Andrew Hogue, Google, Jan. 2011 (1hr, 19min)
- Building Software at Google Scale by Michael Barnathan et al March, 2012 (1hr, 10min)
- Large-scale data analysis using the app engine pipeline API by Brett Slatkin, May 2011, (51min)
- JavaScript Testing at Google Scale by Cory Smith, August, 2011, (1hr, 7min)
- Doubleclick Ad Exchange by E. Manor et al, August, 2012, (1hr,20min)
- How Google Makes Improvements to its search algorithm August 2011, (0:4min)
- How Google's Algorithm Works, Dec 2012, (0:4min)

Google Videos Focusing on Women in the Workplace

- Kathi Pham, Falling in Love with CS (8 min)
- Pavni Diwanji, VP of Engineering (1 min)
- Elena Kon, Embracing Failure (7 min)
- Women Tech Makers Rani Parchuri (5 min)
- Women Tech Makers Hariny Murli (6 min)
- Product Management in Tech Gayathri Rajan Episode 2 (6 min)
- Product Management in Tech, Aparna Chennapragade, Episode 3 (7 min)
- Product Management in Tech, Shimrit Ben-Yair Episode 4 (11 min)

Other items of interest Google Algorithm Change History

lynda.com is a website that develops online software training, or as they adver tise it "videos that really work". USC has paid for the entire campus to have f ree access to these videos. Many of the videos have to do with various aspects of Web Development. Many of the videos are quite good.

go to
http://www.usc.edu/its/lynda/
and login. It may be possible to do so from off campus as well as on-campus as the USC Shibboleth prompts for you USC login and password.

Their entire list of videos can be found at http://www.lynda.com/Web-Interactive-training-tutorials/88-0.html
Below is a selection of videos that apply directly to this course.

- **(Start by logging in to www.usc.edu/its/lynda)**
  - SEO Fundamentals
  - Improving SEO Using Accessibility Techniques
  - Search Engine Optimization Getting Started (2010)
  - Analyzing Your Website to Improve SEO
  - Flash CS4 Professional: Building Search Engine Friendly Sites
  - SEO: Link Building in Depth

Course Assignments

- **There will be five programming assignments**
- **Deliverables**
  - **a short write-up of your solution to the assignment, and**
  - **any source code you have written**
- **Due dates are listed on the *Assignments* page and the *Schedule of Lectures* page**

Copyright Ellis Horowitz, 2011 - 2022

Course Materials

- *Introduction to Information Retrieval* by Manning et al

is our main textbook;

- The book is available for free via PDF download
- It presents all of the classic Information Retrieval concepts

Class NewsGroup Piazza

CSCI 572 Home Page

Not Secure | csci572.com

Apps | CSCI 572 Home P... | CSCI 571 - Home | Computer Science... | ITS - Software | Masters Student... | University of Sout... | Other Bookmarks

- Home Page
- Recent Search Engine Articles
- Schedule of Lectures
- Assignments
- Special Resources
- Course Grading
- Course Materials
- Class News Group Piazza
- Your Grades
- Academic Integrity Policy

Class NewsGroup: Piazza

Piazza is a next generation Question & Answer system specifically designed to get you answers fast.

The quicker you begin asking questions on Piazza (rather than individual email to a classmate or to one of us), the quicker you'll start to reap the benefits of the collective knowledge of your fellow classmates and instructor. I encourage you to ask questions when you're struggling to understand a concept, Piazza lets you participate anonymously too.

Watch their 2 minute introduction video to learn how Piazza is designed to get you answers fast: http://www.piazza.com/signup?video=video-pane, or a not so recent article about them in the New York Times: http://www.nytimes.com/2011/07/04/technology/04piazza.html.

To join piazza you can sign in to the course by yourself using these directions

In this group, you can ask course related and homework related questions. Please make sure to use the class news group to clarify any questions that you may have on the assignments and lab specifications and on the grading guidelines. Please refrain from posting **any code**, which can lead to any sort of plagiarism. The class newsgroup is not the place to debug your code.

More information about Piazza can be found here. If you still have any further questions, please contact the class TAs.

Warning

Piazza has started to sell the information it collects about students to third parties. If you wish to avoid this possibility, when you arrive at the **Set Up Your Piazza Account** dialog box, you will notice the heading **Piazza Careers**. There is a box labeled **Sign me up** and it is already checked.
I suggest you uncheck the box.

Login to Piazza

Once activated, the CSCI 572 newsgroup is accessible at http://www.piazza.com. Click **Login** in the upper right corner. Then enter **Your Email** and **Password** and click **Login**. You should now see the Piazza desktop.

Rules for Using the Class Newsgroup

- Posts to the newsgroup are readable by all so be careful with what you say. There is no need to CC the TA and/or the Professor. All of us will be able to see your message. Students are welcome to respond to questions as well in order to help each other.
- You will be able to post "anonymously" so only the Professor and TA will see your questions.
- Only contact TAs and Producers by sending them direct e-mails in case that your question cannot be answered through this group. If you choose to send TAs an e-mail you must CC all TAs and Producers. Otherwise your e-mail will be ignored.
- **Do not post code and then ask "what's wrong with it, please help"** and expect the audience to fix it for you. The purpose of this group is NOT to debug your code. Such messages will be **deleted**. If you insist on posting code, your membership will be suspended.
- Do not post links to your homework solutions for others to see and help you with a problem you are facing.Such messages will be **deleted**. If you insist on posting code, your membership in the newsgreouo will be suspended.
- Do not post questions about how your homework was graded. Contact the individual grader directly by e-mail.
- Questions about the specifications, the grading guidelines, and allowed software libraries are allowed and welcome.
- Read already existing posts before creating a new discussion. Your question might already have been answered. Questions that have already been answered will be ignored.

If you are not already signed up you can sign up yourself

BEWARE

Review the rules

Academic Integrity Policy

CSCI 572 Home Page

Not Secure | csci572.com

Apps | CSCI 572 Home P... | CSCI 571 - Home | Computer Science... | ITS - Software | Masters Student... | University of Sout... | Other Bookmarks

- Home Page
- Recent Search Engine Articles
- Schedule of Lectures
- Assignments
- Special Resources
- Course Grading
- Course Materials
- Class News Group Piazza
- Your Grades
- Academic Integrity Policy

Course Academic Integrity Policy

Statement on Academic Conduct and Support Systems

Academic Conduct Plagiarism - presenting someone else.s ideas as your own, either verbatim or recast in your own words - is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in SCampus in Section 11, Behavior Violating University Standards https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriate-sanctions/ Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct/. Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the Office of Equity and Diversity http://equity.usc.edu/ or to the Department of Public Safety http://capsnet.usc.edu/department/department-public-safety/online-forms/contact-us. This is important for the safety whole USC community. Another member of the university community - such as a friend, classmate, advisor, or faculty member - can help initiate the report, or can initiate the report on behalf of another person. The Center for Women and Men http://www.usc.edu/student-affairs/cwm/ provides 24/7 confidential support, and the sexual assault resource center webpage sarc@usc.edu describes reporting options and other resources.

Support Systems

A number of USC.s schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the American Language Institute http://dornsife.usc.edu/ali which sponsors courses and workshops specifically for international graduate students. The Office of Disability Services and Programs http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, USC Emergency Information http://emergency.usc.edu/ will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.

Statement on Diversity

The diversity of the participants in this course is a valuable source of ideas, problem solving strategies, and engineering creativity. We encourage and support the efforts of all of our students to contribute freely and enthusiastically. We are members of an academic community where it is our shared responsibility to cultivate a climate where all students and individuals are valued and where both they and their ideas are treated with respect, regardless of their differences, visible or invisible.

Honor Code

Engineering enables and empowers our ambitions and is integral to our identities. In the Viterbi community, accountability is reflected in all our endeavors.

Engineering+ Integrity.
Engineering+ Responsibility.
Engineering+ Community.
Think good. Do better. Be great.

These are the pillars we stand upon as we address the challenges of society and enrich lives.

Public Cloud Source Code Repositories

Some students have been posting the code for CSCI 572 exercises on Github's public repository. Repositories like GitHub and Bitbucket are crawled by search engines. Thus they are easily found and copied. Anyone putting their source code in a public repository is playing with fire and putting their grade at risk. **Public repositories like GitHub and Bit bucket should not be used to store CSCI 572 homework source code.** Private repositories from the same services can be used.

Is this Course Just About Google?

- Yes
  - Google has been innovating in search engines so it makes sense to study what they have done
  - Google encourages their employees to publish the results of their work, so the information is available
- No
  - Many topics in the course are generic to search engines, e.g. crawling, de-duplication, stemming, inverted indexes, etc

Copyright Ellis Horowitz, 2011 - 2022

So…today

- **Quick lecture on search engine history and usage**

Copyright Ellis Horowitz, 2011 - 2022

Once Again

- **All video lectures are available to everyone at https://courses.uscden.net/**
- **Email me if you wish to attend my recording of the lecture**
- **The two exams will be open book and online and are scheduled for:**
  - **Exam 1, Sept 29th**
  - **Exam 2, Dec 1st**
  - ***POTENTIAL PROBLEM*: csci571 has exams the SAME DAY at 6:00-6:40pm**
- **Email me if you wish to attend my taping session**
- **All slide lectures are available to everyone at the website:**

  Login/password for the slides are: csci572/notes
- **REMINDER: Re-Grading of Exams will only permitted for those people scoring 5 or more points below the average**

Please do not take this class if you do not agree with this policy

Copyright Ellis Horowitz, 2011 - 2022

Search Engine Evaluation

Copyright Ellis Horowitz 2011-2022

Outline

- Defining precision/recall
- Mean Average Precision
- Harmonic Mean and F Measure
- Discounted Cumulative Gain
- Elements of Good Search Results
- Google's Search Quality Guidelines
- Using log files for evaluation
- A/B Testing

Copyright Ellis Horowitz, 2011-2022

Comparing Bing and Google

- This site is no longer active, but we can simulate the experiment

Try it yourself!

here are some queries:

- ac versus dc current
- best bottled water
- worst hotel in Santa Monica
- how many gears do I need on a bicycle
- Clint Eastwood's best movie

Copyright Ellis Horowitz 2011-2022

Precision/Recall

- How do we measure the quality of search engines?
- Precision = #(relevant items retrieved)
  divided by
  #(all retrieved items)

- Recall = #(relevant items retrieved)
  divided by
  #(all relevant items)

Copyright Ellis Horowitz, 2011-2022

Formalizing Precision/Recall

	Relevant	Nonrelevant
Retrieved	True positive (tp)	False positive (fp)
Not retrieved	False negative (fn)	True negative (tn)

$$\text{Precision} = tp/(tp + fp)$$

$$\text{Recall} = tp/(tp + fn)$$

- The accuracy of an engine is defined as:

the fraction of these classifications that are correct

$$(tp + tn) / (tp + fp + fn + tn)$$

For web applications,
Precision is more important than Recall

Copyright Ellis Horowitz, 2011-2022

Precision/Recall Using Set Notation

A is set of relevant documents,
B is set of retrieved documents

	Relevant	Non-Relevant
Retrieved	$A \cap B$	$\overline{A} \cap B$
Not Retrieved	$A \cap \overline{B}$	$\overline{A} \cap \overline{B}$

you may not be able to see them, but A and B have a bar over them and it denotes the complement set

$$Recall = \frac{|A \cap B|}{|A|}$$

$$Precision = \frac{|A \cap B|}{|B|}$$

https://en.wikipedia.org/wiki/Precision_and_recall

Copyright Ellis Horowitz, 2011-2022

Precision/Recall Two Observations

- **You can get high recall (but low precision) by retrieving all docs for all queries!**
  - a rather foolish strategy
- **In a good system, precision decreases as the number of docs retrieved (or recall) increases**
  - This is not a theorem, but a result with strong empirical confirmation
  - E.g. viewing multiple pages of Google results often does not improve precision at all

Copyright Ellis Horowitz, 2011-2022

Harmonic Mean

- **There are three Pythagorean means**
  - 1. *arithmetic mean*, 2. *geometric mean*, 3. *harmonic mean*
  - of course we all know how to compute the arithmetic mean
  - the geometric mean is the *n*th root of the product of *n* numbers
- **The harmonic mean tends strongly toward the least element of the list making it useful in analyzing search engine results**
- **To find the harmonic mean of a set of *n* numbers**
  1. add the reciprocals of the numbers in the set
  2. divide the sum by *n*
  3. take the reciprocal of the result
- **e.g. for the numbers 3, 6, 9, and 12**
  - **The arithmetic mean is: (3+6+9+12)/4 = 7.5**
  - **The geometric mean is: nth-root(3*6*9*12) = 4th-root(1944) = 6.64**
  - **The harmonic mean is: (1/3+1/6+1/9+1/12)=(.33+.16+.11+.08)/4=0.17 and 1/0.17 = 5.88**

Copyright Ellis Horowitz, 2011-2022

F Measure

- The harmonic mean of the precision and the recall is often used as an aggregated performance score for the evaluation of algorithms and systems: called the

  F-score (or F-measure).
- *Harmonic mean* of recall and precision is defined as

$$F = \frac{1}{\frac{1}{2}(\frac{1}{R}+\frac{1}{P})} = \frac{2RP}{(R+P)}$$

  - harmonic mean emphasizes the importance of small values, whereas the arithmetic mean is affected more by outliers that are unusually large
- More general form of F-Measure
  - β is a parameter that controls the relative importance of recall and precision

$$F_\beta = (\beta^2+1)RP/(R+\beta^2 P)$$

Copyright Ellis Horowitz, 2011-2022

Calculating Recall/Precision at Fixed Positions

= the relevant documents

Steps

Recall: 1/6 = 0.17
Precision: 1/1

Recall: 1/6 = 0.17
Precision: 1/2 = 0.5

Recall: 2/6 = 0.33
Precision: 2/3 = 0.67

Recall: 3/6 = 0.5
Precision: 3/4 = 0.75

Ranking #1

Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6

e.g. Google Result

Ranking #2

Recall	0.0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6

e.g. Bing Result

Recall=#RelevItemsRetr/allRelevItems
Prec=#RelevItemsRetr/allItemsRetr

Copyright Ellis Horowitz 2011-2022

Average Precision of the Relevant Documents

= the relevant documents

computes the sum of the precisions of the relevant documents

Ranking #1

Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6

Ranking #2

Recall	0.0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6

$$\text{Ranking \#1: } (1.0 + 0.67 + 0.75 + 0.8 + 0.83 + 0.6)/6 = 0.78$$

$$\text{Ranking \#2: } (0.5 + 0.4 + 0.5 + 0.57 + 0.56 + 0.6)/6 = 0.52$$

Conclusion: Ranking #1 for this query is best

Copyright Ellis Horowitz, 2011-2022

Averaging Across Queries

= relevant documents for query 1

Ranking #1

Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

= relevant documents for query 2

Ranking #2

Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3

Average precision across the two queries for relevant docs is:
(1 + .67 + .5 + .44 + .5 + .5 + .4 + .43)/8 = 0.55

Copyright Ellis Horowitz, 2011-2022

Averaging Across Queries

- ***Mean average precision (MAP)*** for a set of queries is the mean of the average precision scores for each query.

$$\mathrm{MAP} = \frac{\sum_{q=1}^{Q} \mathrm{AveP}(q)}{Q}$$

where *Q* is the number of queries

- **Summarize rankings from multiple queries by averaging average precision**
- **This is the *most commonly used* measure in research papers**
- **Assumes user is interested in finding many relevant documents for each query**
- **Requires many relevance judgments in text collection**

Copyright Ellis Horowitz, 2011-2022

Mean Average Precision Example

= relevant documents for query 1

Ranking #1

Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

= relevant documents for query 2

Ranking #2

Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3

$$average\ precision\ query\ 1 = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$$
$$average\ precision\ query\ 2 = (0.5 + 0.4 + 0.43)/3 = 0.44$$

$$mean\ average\ precision = (0.62 + 0.44)/2 = 0.53$$

Copyright Ellis Horowitz 2011-2022

More on Mean Average Precision Calculation

- **Mean Average Precision (MAP)**
  - Some negative aspects
  - If a relevant document never gets retrieved, we assume the precision corresponding to that relevant doc to be zero (this is actually reasonable)
  - Each query counts equally
  - MAP assumes user is interested in finding many relevant documents for each query
  - MAP requires many relevance judgments in the document collection

Difficulties in Using Precision/Recall

- **Should average over large document collection and query ensembles**
- **Need human relevance assessments**
  - But people aren't always reliable assessors
- **Assessments have to be binary**
  - Nuanced assessments?
- **Heavily skewed by collection/authorship**
  - Results may not translate from one domain to another

Copyright Ellis Horowitz, 2011-2022

A Final Evaluation Measure: Discounted Cumulative Gain

- **The premise of DCG is that highly relevant documents appearing lower in a search result list should be penalized as the graded relevance value is reduced logarithmically proportional to the position of the result.**
- **The discounted CG accumulated at a particular rank position *p* is defined as**

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2(i+1)}$$

where rel_i is the graded relevance of the result at position *i*

- **Gain is accumulated starting at the top of the ranking and may be reduced, or *discounted*, at lower ranks**
- **Typical discount is 1/log *(rank)***
  - **With base 2, the discount at rank 4 is 1/2, and at rank 8 it is 1/3**
- **An alternative formulation of DCG places stronger emphasis on retrieving relevant documents:**

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

Copyright Ellis Horowitz, 2011-2022

Discounted Cumulative Gain Example

we want high weights for high rank documents, because searchers are likely to inspect them, and low weights for low rank documents that searchers are unlikely to ever see.

The discount factor is commonly chosen as log2(rank + 1) and is used to divide the relevance grade.

Using a logarithm for the position penalty makes the decay effect more gradual compared to using the position itself.

Discount examples

Rank	Grade	Discount1 [1/rank]	Discount2 [log2(rank + 1)]	Discount1 Grade	Discount2 Grade
1	4	1.000	1.000	4.000	4.000
2	3	0.500	0.631	1.500	1.893
3	2	0.333	0.500	0.667	1.000
4	1	0.250	0.431	0.250	0.431
5	1	0.200	0.387	0.200	0.387

Search Engine Evaluation Metrics

Metrics table

Scale	Metric	Measures	Drawbacks
Binary	Precision (P)	The relevance of the entire results set (gridded results display)	Doesn't account for position
Binary	Average Precision (AP)	Relevance to a user scanning results sequentially	Large impact of low-rank results
Graded	Cumulative Gain (CG)	Information gain from a results set	Same as *Precision* doesn't factor in position
Graded	Discount Cumulative Gain (DCG)	Information gain with positional weighting	Difficult to compare across queries
Graded	normalized DCG (nDCG)	How close the results are to the best possible	No longer shows information gain

Finally see https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval)

Copyright Ellis Horowitz 2011-2022

How Evaluation is Done at Web Search Engines

- Search engines have test collections of queries and hand-ranked results
- Recall is difficult to measure on the web
- Search engines often use precision at top *k* positions, e.g., *k = 10*
- . . . or measures that reward you more for getting rank *1* right than for getting rank *10* right.
- Search engines also use non-relevance-based measures
  - Click-through on first result
    - Not very reliable if you look at a single click-through …

    but pretty reliable in the aggregate.
  - Studies of user behavior in the lab
  - A/B testing

Google's Search Quality Rating Guidelines Document

- **Google relies on raters, working in many countries and languages around the world**
- **The data they generate is rolled up statistically to give**
  - a view of the quality of search results and search experience over time, and
  - an ability to measure the effect of proposed changes to Google's search algorithms

General Guidelines

October 14, 2020

General Guidelines Overview 5
Introduction to Search Quality Rating 6
0.0 The Search Experience 6
0.1 The Purpose of Search Quality Rating 6
0.2 Raters Must Represent People in their Rating Locale 6
0.3 Browser Requirements 7
0.4 Ad Blocking Extensions 7
0.5 Internet Safety Information 7
0.6 The Role of Examples in these Guidelines 7
Part 1: Page Quality Rating Guideline 8
1.0 Introduction to Page Quality Rating 8
2.0 Understanding Webpages and Websites 8
2.1 Important Definitions 8
2.2 What is the Purpose of a Webpage? 9
2.3 Your Money or Your Life (YMYL) Pages 10
2.4 Understanding Webpage Content 10
2.4.1 Identifying the Main Content (MC) 10
2.4.2 Identifying the Supplementary Content (SC) 11
2.4.3 Identifying Advertisements/Monetization (Ads) 11
2.4.4 Summary of the Parts of the Page 12
2.5 Understanding the Website 12
2.5.1 Finding the Homepage 12
2.5.2 Finding Who is Responsible for the Website and Who Created the Content on the Page 14
2.5.3 Finding About Us, Contact Information, and Customer Service Information 14
2.6 Reputation of the Website or Creator of the Main Content 15
2.6.1 Research on the Reputation of the Website or Creator of the Main Content 16
2.6.2 Sources of Reputation Information 16
2.6.3 Customer Reviews of Stores/Businesses 16
2.6.4 How to Search for Reputation Information 16
2.6.5 What to Do When You Find No Reputation Information 18
3.0 Overall Page Quality Rating 19
3.1 Page Quality Rating: Most Important Factors 19
3.2 Expertise, Authoritativeness, and Trustworthiness (E-A-T) 19
4.0 High Quality Pages 20
4.1 Characteristics of High Quality Pages 20
4.2 A Satisfying Amount of High Quality Main Content 21
4.3 Clear and Satisfying Website Information: Who is Responsible and Customer Service 21
4.4 Positive Reputation 21
4.5 A High Level of Expertise/Authoritativeness/Trustworthiness (E-A-T) 22
4.6 Examples of High Quality Pages 22
5.0 Highest Quality Pages 26

http://csci572.com/papers/2020_10searchqualityevaluatorguidelines.pdf

Copyright Ellis Horowitz, 2011-2022

Google's Search Quality Ratings Guidelines Document

- This document gives evaluators examples and guidelines for appropriate ratings.
- the evaluator looks at a search query and a result that could be returned. They rate the relevance of the result for that query on a scale described within the document.

The six rating scale categories

searchqualityevaluatorgui

www-scf.usc.edu/~csci572/papers/searchqualityevaluatorguidelines.pdf

Apps | Bookmarks | CSCI 571 Home Page | CSCI 572 Home Page | Computer Science De | Other bookmarks

Rating Scale	Description
Vital	A special rating category. See Section 4.1 of the Rating Guidelines.
Useful	A page that is very helpful for most users.
Relevant	A page that is helpful for many or some users.
Slightly Relevant	A page that is not very helpful for most users, but is somewhat related to the query. Some or few users would find this page helpful.
Off-Topic or Useless	A page that is helpful for very few or no users.
Unratable	A page that cannot be evaluated. A complete description can be found in Section 4.6 of the Rating Guidelines.

Copyright Ellis Horowitz, 2011-2022

Google's 4-Step Process for Changing Their Search Algorithm

1. Precision Evaluations
People use the Guidelines to rate search results

2. Side-by-Side Experiments
people are shown two different sets of search results and asked which they prefer

3. Live Traffic Experiments
the search algorithm is altered for a small number of actual users

4. Full Launch
A final analysis by Google engineers and the improvement is released

Copyright Ellis Horowitz 2011-2022

A/B Testing at Web Search Engines

- **A/B testing** is comparing two versions of a web page to see which one performs better. You compare two web pages by showing the two variants (let's call them **A** and **B**) to similar visitors at the same time. The one that gives a better conversion rate, wins!

1. **Purpose:** Test a single innovation
2. **Prerequisite:** You have a large search engine up and running.
3. **Have most users use old system**
4. **Divert a small proportion of traffic (e.g., 1%) to an experiment to evaluate an innovation**
5. **Evaluate with an automatic measure like click through on first result**

- we directly see if the innovation does improve user happiness
- *This is the evaluation methodology large search engines trust the most*

Copyright Ellis Horowitz, 2011-2022

USING USER CLICKS FOR EVALUATION

What Do Clicks Tell Us?

There is strong position bias, so absolute click rates unreliable

Copyright Ellis Horowitz, 2011-2022

Relative vs Absolute Ratings

ALL RESULTS

RELATED SEARCHES
CIKM 2008

SEARCH HISTORY
Turn on search history to start remembering your searches.
Turn history on

ALL RESULTS — 1-10 of 131,000 results · Advanced

CIKM 2008 | Home
Napa Valley Marriott Hotel & Spa: Napa Valley, California October 26-30, 2008
cikm2008.org · Cached page

Papers · Program Committee · Themes · News · Important Dates · Napa Valley · Banquet · Posters
Show more results from cikm2008.org

Conference on Information and Knowledge Management (CIKM)
Provides an international forum for presentation and discussion of research on information and knowledge management, as well as recent advances on data and knowledge bases ...
www.**cikm**.org · Cached page

Conference on Information and Knowledge Management (CIKM'02)
SAIC Headquarters, McLean, Virginia, USA, 4-9 November 2002.
www.**cikm**.org/2002 · Cached page

ACM **CIKM** 2007 - Lisbon, Portugal
News and announcements: 12/02 - Best interdisciplinary paper award at **CIKM** 2007 went to Fei Wu and Daniel Weld for Autonomously Semantifying Wikipedia.
www.fc.ul.pt/**cikm**2007 · Cached page

CIKM 2009 | Home
CIKM 2009 (The 18th ACM **Conference on Information and Knowledge Management**) will be held on November 2-6, 2009, Hong Kong. Since 1992, **CIKM** has successfully brought together ...
www.comp.polyu.edu.hk/conference/**cikm**2009 · Cached page

Conference on Information and Knowledge Management (CIKM)
CIKM Conference on Information and Knowledge Management The **Conference on Information and Knowledge Management (CIKM)** provides an international forum for presentation and ...
cikmconference.org · Cached page

User's click sequence

Hard to conclude Result1 > Result3
Probably can conclude Result3 > Result2

Copyright Ellis Horowitz, 2011-2022

Query Logs

- **Used for both tuning and evaluating search engines**
  - also for various techniques such as query suggestion
- **Typical contents of the query log files**
  - User identifier or user session identifier
  - Query terms - stored exactly as user entered them
  - List of URLs of results, their ranks on the result list, and whether they were clicked on
  - Timestamp(s) - records the time of user events such as query submission, clicks

Copyright Ellis Horowitz, 2011-2022

How Query Logs Can Be Used

- **Clicks are not relevance judgments**
  - although they are correlated
  - biased by a number of factors such as rank on result list
- **Can use clickthough data to predict *preferences* between pairs of documents**
  - appropriate for tasks with multiple levels of relevance, focused on user relevance
  - various "policies" used to generate preferences

Copyright Ellis Horowitz, 2011-2022

A Final Thought
Google's Enhancements of Search Results

Display improvements
- immediate answers
- autocomplete anticipations

Extensions to More Data
- results from books
- results from news
- results from images
- results from patents
- results from air schedules

New Input forms
- search by voice
- search by image

information retrieval improvements
- snippets
- spelling correction
- translations
- People Also Ask boxes
- use of synonyms
- use of knowledge graph

The page below discusses the many aspects that go into producing search results at Google

https://www.google.com/search/howsearchworks

Copyright Ellis Horowitz, 2011-2022

Final Thought

- See wikipedia on
- https://en.wikipedia.org/wiki/Comparison_of_web_search_engines

Copyright Ellis Horowitz, 2011-2022

Web Trends and Measurements

Copyright © Ellis Horowitz 1999-2022

Web Trends

- Web has changed dramatically over the last 30+ years
- If one is building a web search engine today it is important to understand the different dimensions of the web
  - Scale, complexity and growth are only a few important factors
- In today's lecture I try to quantify some of the trends to better understand where the web is going and why
- many of the early slides come from Mary Meeker (formerly of) Kleiner, Perkins, Caufield and Byers, a major venture capital firm, http://www.kpcb.com/

Copyright © Ellis Horowitz 1999-2022

A total of **5.03 billion** people around the world use the internet today – equivalent to 63.1 percent of the world's total population

The US leads in the development of highly popular Internet websites;

Baidu is a Chinese search engine

Tencent is a Chinese holding company of Internet properties, among the most popular being, QQ, for chatting;
***Sohu**.com Inc. is a Chinese online media, search, gaming, community and mobile service group.*

3/14 – 6 of Top 10 Global Internet Properties 'Made in USA'... >86% of Their Users Outside America...China Rising Fast

@KPCB Source: comScore, 3/14.

131

China Mobile Internet Users = 817MM.+9% vs. +8% Y/Y

Copyright © Ellis Horowitz 1999-2022

China Mobile Internet Usage Leaders...
Tencent + Alibaba + Baidu = 71% of Mobile Time Spent

Share of Mobile Time Spent, April 2016
Daily Mobile Time Spent = ~200 Minutes per User, Average

Tencent
- WeChat
- QQ
- QQ Browser
- Tencent Video
- Tencent News
- Tencent Games
- QQ Music
- JD.com
- QQ Reading

Alibaba
- UCWeb Browser
- Taobao
- Weibo
- YouKu Video
- Momo
- Shuqi Novel
- AliPay
- AutoNavi

Baidu
- Mobile Baidu
- iQiyi / PPS Video
- Baidu Browser
- Baidu Tieba
- 91 Desktop
- Baidu Maps

- All Other

@KPCB

Note: Grouping of apps include strategic investments made by Tencent, Alibaba and Baidu. Only apps in top 50 by time spent share are called out. Source: QuestMobile, Trustdata, and Hillhouse estimates.

Hillhouse Capital

Global smartphone shipments from 2009 to 2021

Details: Worldwide; IDC; 2009 to 2021

© Statista 2022

World's Content is Increasingly Findable + Shared + Tagged - Digital Info Created + Shared up 9x in Five Years

There has been exponential growth in online information;
1 Zettabyte = 1,024 Exabytes
1 Exabyte = 1,024 Petabytes
1 Petabyte = 1,024 Terabytes
1 Terabyte = 1,024 Gigabytes
or
1 Zettabyte = 1,000,000,000,000 gigabytes

Amount of global digital information created & shared – from documents to pictures to tweets – grew 9x in five years to nearly 2 zettabytes in 2011, per IDC.*

KPCB

Note: * 1 zettabyte = 1 trillion gigabytes. Source: IDC report "Extracting Value from Chaos" 6/11.

Photos Alone = 1.8B+ Uploaded & Shared Per Day...
Growth Remains Robust as New Real-Time Platforms Emerge

500 million photos are uploaded every day and that number is doubling every year

Yahoo has recently made a major upgrade to **Flickr**

Instagram was in 2010 purchased by Facebook for $1 billion

Snapchat is a photo messaging application developed by two Stanford students ($9B valuation);

bobby Murphy - Evan Spiegel

@KPCB Source: KPCB estimates based on publicly disclosed company data, 2014 YTD data per latest as of 5/14.

62

Content Uploaded to YouTube

Internet › Online Video & Entertainment

Hours of video uploaded to YouTube every minute as of February 2020

- As of February 2020, more than 500 hours of video were uploaded to YouTube every minute.
- This equates to approximately 30,000 hours of newly uploaded content per hour.
- The number of video content hours uploaded every 60 seconds grew by around 40 percent between 2014 and 2020.

Mobile Accounts for 50% of Web Traffic

In the second quarter of 2022, mobile devices (excluding tablets) generated 58.99 percent of global website traffic, consistently hovering around the 50 percent mark since the beginning of 2017 before permanently surpassing it in 2020.

Tablet Growth =
More Rapid than Smartphones, iPad = ~3x iPhone Growth

KPCB

Source: Apple, as of CQ1:13 (12 quarters post iPad launch).
Launch Dates: iPhone (6/29/07), iPad (4/3/10).

44

Copyright © Ellis Horowitz 2021

Product Finding = Often Starts @ Search (Amazon + Google...)

Where Do You Begin Your Product Search?

Source: Survata (9/17). Note: n = 2,000 USA customers.

KLEINER PERKINS
2018
INTERNET TRENDS

Copyright © Ellis Horowitz 1999-2022

Technology Cycles – Still Early Cycle on Smartphones + Tablets, Now Wearables Coming on Strong, Faster than Typical 10-Year Cycle

Technology Cycles Have Tended to Last Ten Years

Mainframe Computing 1960s	*Mini* Computing 1970s	*Personal* Computing 1980s	*Desktop Internet* Computing 1990s	*Mobile Internet* Computing 2000s	*Wearable / Everywhere* Computing 2014+

Others?

KPCB

49

Image Source: Computersciencelab.com, Wikipedia, IBM, Apple, Google, NTT docomo, Google, Jawbone, Pebble.

Re-Imagination of Computing Operating Systems - iOS + Android = 60% Share vs. 35% for Windows

Global Market Share of Personal Computing Platforms by Operating System Shipments, 1975 – 2012

KPCB

Source: Asymco.com (as of 2011), Public Filings, Morgan Stanley Research, Gartner for 2012 data.

Major Cloud Providers

Market Share Of Leading Cloud Hosting Providers

Top 10 Providers by Total 2020 Market Share

Source: Intricately data, April 2021

...While The Cloud Rises

@KPCB *Note: S3 is AWS' storage product and used as proxy for AWS scale / growth .
Source: Company data.

74

Copyright © Ellis Horowitz 1999-2022

...Computing Big Bangs Volume Effects = Cloud Revenue Re-Accelerating +58% vs. +54% Q/Q

Cloud Service Revenue – Amazon + Microsoft + Google

KLEINER PERKINS
2018
INTERNET TRENDS

Source: Amazon AWS = Company filings, Microsoft Azure = Keith Weiss @ Morgan Stanley (4/18), Google Cloud = Brian Nowak @ Morgan Stanley (5/18). Note: Google Cloud revenue excluded in Y/Y growth rate calculation due to limited quarterly estimates.

185

Copyright © Ellis Horowitz & Marco Papa 1999-2022

Google Voice Search Queries

Google Trends imply queries associated with voice-related commands have risen >35x since 2008 after launch of iPhone & Google Voice Search

Google Trends, Worldwide, 2008 – 2019

Copyright Ellis Horowitz, 2011-2022

Voice-Based *Mobile* Platform *Front-Ends* = Voice Can Replace Typing

Google Assistant

Nearly 70% of Requests are Natural / Conversational Language, 5/17

20% of Mobile Queries Made via Voice, 5/16

KLEINER PERKINS

Source: Google I/O (5/16), Image: Macrumors (2/17)

Copyright © Ellis Horowitz 1999-2022

.Voice =
47MM Amazon Echo Base + ~2x in One Year

Amazon Echo Installed Base

Amazon Echo Skills

BOND
Internet Trends
2019

Source: Consumer Intelligence Research Partners LLC (Echo install base, 2/18), Various media outlets including Geekwire, TechCrunch, & Wired (Echo skills, 4/19)

51

Copyright Ellis Horowitz, 2011-2022

21

Summary of Recent Trends in Web/Internet Development

- Growth in number of users connected
- Growth in Smartphone use
- Growth in digital data, especially photos and video
- Growth in Social Media as an advertising platform
- Transition from desktop/laptop use to mobile
- Growth in tablet usage over desktops/laptops
- Decreased dominance of Microsoft Windows
- Move away from server farms to cloud computing
- Growth in voice communication with devices

Copyright © Ellis Horowitz 1999-2021

Measuring the Web

- The World Wide Web (the Web, the publicly accessible web) is so dynamic it is hard to describe it and have the description be valid for very long
- In this lecture we look at what is known,
  - Measuring the Web by number of web sites
  - Measuring the Web by the Languages of Web Pages
  - Measuring the Web by Rate of Change of Pages
  - Measuring the Web by Document Content Type
  - Measuring the Web by linkage
  - Measuring the Web as a Graph
  - Measuring the Web by Content
  - (using the best statistics we can find)

Copyright Ellis Horowitz, 2011-2022

Number of Websites

Jan. 2020:
~1.7 Billion sites

nginx web server had the largest growth;

Over 50% of websites Are hosted either by Apache or nginx;

But Microsoft web servers still power 43.2% of all sites

Around 75% of websites are not active, but parked

http://www.internetlivestats.com/total-number-of-websites/

Copyright © Ellis Horowitz 1999-2022

Number of websites in the world

The global number of websites has more than doubled from 2015 to 2021. Websites growth rate from 1991 to 2021

Distribution Across TLDs

136 million in .com, 21million in .tk, 14 million in .de, etc

what is .tk and why is it so large? (Tokelau)

Domain Count Statistics for TLDs

This page displays the count of all Domains in each TLD. For Registry's publishing a domain count, "Our Count" should closely match their published record. For registry's that don't provide a zone file or publish an up-to-date record, Our Count represents all domains we know about, which is usually more accurate.

TLD	Our Count
.com	136,346,743
.tk	21,014,704
.de	14,586,920
.net	14,061,971
.cn	11,579,296
.org	10,398,501
.uk	10,361,314
.info	5,528,696
.nl	5,042,167
.ru	4,928,746
.eu	3,673,059
.br	3,282,608
.fr	3,195,248
.top	2,921,081
.au	2,845,979
.it	2,786,780
.ca	2,611,854
.pl	2,302,199
.biz	2,269,454
.loan	2,253,686

Display: Top25 - Default - None

Web Page Language Diversity

- **The Web contains pages in many different languages**
- Characters in a language are encoded such that each character is paired with a number
- Unicode and its parallel standard, the ISO/IEC 10646 Universal Character Set, together constitute a modern, unified character encoding.
- Most modern web browsers feature automatic character encoding detection. In Firefox, for example, see the View/Character Encoding submenu, shown below
- In HTML one can specify the character encoding using
- **<meta http-equiv="Content-Type content="text/html" charset=utf-8>**

- If charset is missing ISO-8859-1 is taken as the default unless there is a browser setting;
- Websites in non-western languages typically use UTF-8

Measuring Language Diversity

- **It is estimated that about 40,000 different languages have been created by human beings**
- **Only between 6,000-9,000 are still in use**
- **Study done by the United Nations**
  - http://unesdoc.unesco.org/images/0018/001870/187016e.pdf
  - The methodology was to examine the pages on a search engine and attempt to identify the primary language in which the page is written
  - **Conclusions**
    - **From 1996 – 2008 English was predominant, occupying roughly 80% of web pages**
    - **At the same time the number of Internet users who had English as their primary language dropped from 80% to 40%**

Content languages for websites as of 12 March 2014[2]

Complexity of Data Types

Proliferation of Content Types Available

- **By some accounts, there are 16,000 to 51,000 content types***
- **What to do with content types?**
  - **Parse them**
    - **How?** The Apache Tika™ toolkit detects and extracts metadata and text content from various documents using existing parser libraries. Tika unifies these parsers under a single interface. Tika is useful for search engine indexing
    - Extract their text and structure
  - **Index their metadata**
    - Use an indexing technology like
    - Lucene, http://lucene.apache.org/
    - Solr, http://lucene.apache.org/solr/
    - Google Search Appliance (http://www.google.com/enterprise/search/products/gsa.html)
  - **Identify what language they belong to**
    - N-grams

*http://filext.com/ (see if you can name the top 20 file extensions)

Copyright Ellis Horowitz, 2011-2022

Content Types Indexed by Google

Considering the fact that there are thousands of file types of content stored on the web, Google actually indexes only a small number, less than 3 dozen, but they may well constitute far more than a majority of the available content

http://support.google.com/webmasters/bin/answer.py?hl=en&answer=35287

Copyright Ellis Horowitz, 2011-2022

A Summary of Some Web Facts

- **How many websites?** *~1.86 billion*
- **How are they distributed across TLDs or across countries?**
  - *112 million out of 148 million belong to .com or about 72%*
- **How many web pages are there?** *30 trillion unique URLs from Google found in 2012,*

  see **http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html**
- **Which content types hold the most information: HTML, PDF, Word, Excel, PPT, others?**
  - *There are thousands of different content types*
- **How much storage is required to hold a single snapshot of the Web?**
  - *1 trillion web pages at 100K bytes per page requires 100 petabytes*
    - *1 petabyte storage costs under $1,000, so $100,000 of equipment will work*
  - *Google processes 24 petabytes per day, http://en.wikipedia.org/wiki/Petabyte*
  - *The Internet Archive has more than 10 petabytes, http://en.wikipedia.org/wiki/Internet_Archive*
- **What are the languages in which the documents are written?**
  - *According to the Internet Archive, about 55% is in English; other popular languages include: French, German, Spanish and Chinese, also see http://en.wikipedia.org/wiki/Languages_used_on_the_Internet*
- **General properties of the Web graph**
  - *In-degree and out-degree distribution follows a power law*
- **Categories of Content: pornography, spam, mirrors**
  - *Presumably there is a lot of the above, but little concrete data on how much*

Copyright Ellis Horowitz, 2011-2022

Manual Hierarchical Web Taxonomies

- **Yahoo originally used human editors to assemble a large hierarchically structured directory of web pages.**
  - **http://www.yahoo.com/**
  - **Yahoo still retains the hierarchy as seen to the left; under european cars we see categories: regional with 93 matches, business & economy with 79 matches, etc**

Copyright Ellis Horowitz, 2011-2022

Open Directory Project

- **Open Directory Project, known as DMoZ, is an effort to organize the web according to an ontology;**
- **An approach similar to Yahoo's;**
- **Based on the distributed labor of volunteer editors ("net-citizens provide the collective brain").**
- **Used by most other search engines.**
- **Started by Netscape.**
  - **http://www.dmoz.org/**
- **Distributes its data using RDF format**
- **DMOZ shut down in 2016**

Copyright Ellis Horowitz, 2011-2022

Drilling Down By Category

Open Directory - Science - Mozilla Firefox

www.dmoz.org/Science/

dmoz open directory project — In partnership with AOL Search.

about dmoz | dmoz blog | report abuse/spam | help

Search — the entire directory

Top: **Science** *(104,480)* — Description

[A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z]

- Agriculture *(3,670)*
- Anomalies and Alternative Science *(480)*
- Astronomy *(3,595)*
- Biology *(31,012)*
- Chemistry *(4,171)*
- Computer Science@ *(1,971)*
- Earth Sciences *(6,098)*
- Environment *(6,572)*
- Math *(9,541)*
- Physics *(4,281)*
- Science in Society *(680)*
- Social Sciences *(19,489)*
- Technology *(10,560)*
- Women@ *(153)*

- Academic Departments *(9)*
- By Region *(0)*
- Chats and Forums *(16)*
- Directories *(27)*
- Educational Resources *(352)*
- Employment *(69)*
- Events *(54)*
- History@ *(346)*
- Instruments and Supplies *(2,370)*
- Libraries@ *(88)*
- Methods and Techniques *(190)*
- Museums@ *(479)*
- News and Media *(234)*
- Organizations *(131)*
- People *(0)*
- Publications *(241)*
- Reference *(389)*
- Research Groups and Centers *(57)*
- Search Engines *(9)*
- Software *(183)*
- Weblogs@ *(117)*

Selecting Category "Science"

Open Directory - Computers: Computer Science - Mozilla Firefox

www.dmoz.org/Computers/Computer_Science/

dmoz open directory project — In partnership with AOL Search.

about dmoz | dmoz blog | report abuse/spam | help

Search — the entire directory

Top: **Computers**: **Computer Science** *(1,971)* — Description

- Academic Departments *(553)*
- Conferences *(203)*
- Directories *(8)*
- Organizations *(71)*
- People *(271)*
- Publications *(80)*
- Reference *(4)*
- Research Institutes *(74)*

- Artificial Intelligence@ *(1,294)*
- Artificial Life@ *(230)*
- Computational Geometry@ *(60)*
- Computer Graphics *(39)*
- Database Theory *(82)*
- Distributed Computing *(225)*
- Parallel Computing@ *(367)*
- Software Engineering@ *(114)*
- Theoretical *(361)*

See also:

- Computers: Algorithms *(284)*
- Computers: Programming *(15,619)*
- Science: Math *(9,541)*
- Science: Technology: Electrical Engineering *(237)*

This category in other languages:

Selecting Category "Computer Science"

Copyright © Ellis Horowitz 1999-2022

Internet Archive

- The Internet Archive has been taking a snapshot of the World Wide Web every two months since 1997 – has used **Apache Nutch**
- The results are made available through the Wayback Machine,
- Its database is approximately 4.5 petabytes
- The founder is Brewster Kahle
- For the past 13 years, the Internet Archive has been growing rapidly, most recently by about 100TB of data per month.
- Their crawler surveys the web every two months The algorithm first performs a broad crawl that starts with a few "seed sites," such as Yahoo's directory. After snapping a shot of the home page, it then moves to any referable pages within the site until there are no more pages to capture. If there are any links on those pages, the algorithm automatically opens them and archives that content as well.

Copyright Ellis Horowitz, 2011-2022

Surface Web

SURFACE WEB
Google
Bing
Wikipedia
DEEP WEB
Academic Information
Medical Records
Legal Documents
Scientific Reports
Subscription Information
Contains 90% of the information on the Internet, but is not accessible by Surface Web crawlers.
Social Media
Multilingual Databases
Financial Records
Government Resources
Competitor Websites
Organization-specific Repositories
(DARK WEB)
A part of the Deep Web accessible only through certain browsers such as Tor designed to ensure anonymity. Deep Web Technologies has zero involvement with the Dark Web.
Illegal Information
Drug Trafficking sites
TOR-Encrypted sites
Political Protests
Private Communications

Search Engine History and Basics

Copyright Ellis Horowitz 2013-2022

A Brief Chronology of Search Engines

- 1991
  - Gopher, Archie, Veronica — early search engines, non-web
- 1993
  - Wanderer,
  - ALIWeb
  - Excite — http://www.excite.com/ — powerful indexing
- 1994
  - Galaxy — http://www.galaxy.com/ — Early searchable directory
  - Yahoo — http://www.yahoo.com/ — Sophisticated searchable directory
  - Lycos — http://www.lycos.com/ — Improved query matching
  - WebCrawler — http://www.webcrawler.com/ — Includes full text of pages
  - Alta Vista — http://www.altavista.com/ — a large index
- 1995
  - Infoseek — http://www.infoseek.com/ — included in Netscape Navigator
  - Metacrawler — http://www.metacrawler.com/ — combines results from other engines
  - SavvySearch — http://www.savvysearch.com/ — combines results from other engines
  - LookSmart — http://www.looksmart.com — convenient organization
- 1996
  - Inktomi — http://www.inktomi.com / — a large index using commodity hardware
  - HotBot, — http://www.hotbot.com/ — a large index
- 1997
  - AskJeeves — http://www.askjeeves.com/ — fancy query processing
- 1998
  - Goto — http://www.goto.com/ — introduces auctioning of positions
  - Google — http://www.google.com — ranking using content and links

- Today there are hundreds of search engines, many are specialized
- See <a href=http://www.searchenginehistory.com/>Search Engine History</a>
- A very long web page describing the history of search engines with lots of good links

Copyright Ellis Horowitz, 2011-2022

Archie, Veronica, Gopher

- **By late 1980's many files were available by anonymous FTP.**
- **In 1990, Alan Emtage, P. Deutsch, et al of McGill Univ. developed Archie (short for "archives")**
  - Assembled lists of files available on many FTP servers.
  - Allowed regex search of these file names.
- **In 1993, Veronica and Jughead were developed to search names of text files available through Gopher servers**
  - The **Gopher protocol** is a TCP/IP application layer protocol designed for distributing, searching, and retrieving documents over the Internet. Strongly oriented towards a menu-document design
  - The Gopher ecosystem is often regarded as the effective predecessor of the World Wide Web

Copyright Ellis Horowitz, 2011-2022

- Excite came from the project Architext, which was started in February, 1993 by six Stanford undergrad students.
  - They had the idea of using statistical analysis of word relationships to make searching more efficient.
  - They were soon funded, and in mid 1993 they released copies of their search software for use on web sites.
- Later developments
  - Excite was bought by a broadband provider named @Home in January, 1999 for $6.5 billion, and was named Excite@Home. In October, 2001 Excite@Home filed for bankruptcy. InfoSpace bought Excite from bankruptcy court for $10 million
  - www.excite.com still exists as a portal

Copyright Ellis Horowitz, 2011-2022

World Wide Web Wanderer

- In June 1993 Matthew Gray while at MIT introduced the World Wide Web Wanderer.
  - Initial goal was to measure the growth of the web by counting active web servers. He soon upgraded the software to capture actual URL's. His database became known as the Wandex.
- The World Wide Web Wanderer was a Perl-based web crawler that was first deployed in June 1993
- Matthew Gray now works for Google.
- While the Wanderer was probably the first web robot, and, with its index, clearly had the potential to become a general-purpose WWW search engine it never went that far
- The Wanderer charted the growth of the web until late 1995.

Copyright Ellis Horowitz, 2011-2022

ALIWEB

- In November of 1993 Martijn Koster created "Archie-Like Indexing of the Web", or ALIWEB in response to the Wanderer.
  - Some consider it to be the first Web search engine
- ALIWEB crawled meta information and allowed users to submit their pages they wanted indexed with their own page description.
- This meant it needed no bot to collect data and was not using excessive bandwidth.
- One downside of ALIWEB was that people did not know how to submit their site

Copyright Ellis Horowitz, 2011-2022

- AltaVista debut online came during December, 1995. AltaVista brought many important features to the web scene.
  - They were the first to allow natural language queries
  - They offered advanced searching techniques
  - They allowed users to add or delete their own URL within 24 hours.
  - They even allowed inbound link checking. AltaVista also provided numerous search tips and advanced search features.
- Later developments
  - On February 18, 2003, Overture signed a letter of intent to buy AltaVista for $80 million in stock and $60 million cash. After Yahoo! bought out Overture they rolled some of the AltaVista technology into Yahoo! Search, and occasionally used AltaVista as a testing platform.

Copyright Ellis Horowitz, 2011-2022

- Lycos was designed at Carnegie Mellon University around July of 1994. Michael Loren Mauldin was responsible for this search engine and was the chief scientist at Lycos Inc in the early years.
- On July 20, 1994, Lycos went public with a catalog of 54,000 documents.
  - In addition to providing ranked relevance retrieval, Lycos provided prefix matching and word proximity bonuses.
  - Lycos' main difference was the sheer size of its catalog: by August 1994, Lycos had identified 394,000 documents; by January 1995, the catalog had reached 1.5 million documents; and by November 1996, Lycos had indexed over 60 million documents -- more than any other Web search engine.
- In October 1994, Lycos ranked first on Netscape's list of search engines
- Lycos has gone through a series of owners, and it still exists as www.lycos.com

Copyright Ellis Horowitz, 2011-2022

- Infoseek also started out in 1994, founded by Steve Kirsch
- In December 1995 they convinced Netscape to use them as their default search engine, which gave them major exposure.
- One popular feature of Infoseek was allowing webmasters to submit a page to the search index in real time, which was a search spammer's paradise
- They were the first search engine to sell advertising on a CPM (Cost per Thousand) impressions basis
- Infoseek was bought by Walt Disney Company in 1998

Copyright Ellis Horowitz, 2011-2022

- In 1994, two Stanford Ph.D. students David Filo and Jerry Yang posted web pages with links on them, organized into a topical hierarchy.
- As the number of links began to grow, they developed a hierarchical listing. As the pages become more popular, they developed a way to search through all of the links.
- Early on all the links on the pages were updated manually rather than automatically by spider or robot and the search feature searched only those links
- Yahoo home page acted as a portal with Email, Finance, and Groups being very successful; however after 2000 usage declined
- After many years of decline Yahoo was purchased by Verizon in 2017 for $4.48 billion, and it lives on

Copyright Ellis Horowitz, 2011-2022

- Looksmart was founded in 1995 in Australia. They competed with the Yahoo! Directory by frequently increasing their inclusion rates
- Later developments
  - In 2002 Looksmart transitioned into a pay per click provider, which charged listed sites a flat fee per click. They syndicated those paid listings to some major portals like MSN.
  - The problem was that Looksmart became too dependant on MSN, and in 2003, when Microsoft announced they were dumping Looksmart that basically killed their business model.
  - In March of 2002, Looksmart bought a search engine by the name of WiseNut, but it never gained traction
- See https://en.wikipedia.org/wiki/LookSmart

Copyright Ellis Horowitz, 2011-2022

- The Inktomi Corporation came about on May 20, 1996 with its search engine Hotbot. Two Cal Berkeley cohorts created Inktomi from the improved technology gained from their research
- Later developments
  - In October of 2001 Inktomi accidentally allowed the public to access their database of spam sites, which listed over 1 million URLs at that time.
  - Inktomi pioneered ***the paid inclusion model*** in which a website pays a fee to the search engine that guarantees the site will be displayed when certain search terms are entered
  - The model was nowhere near as efficient as the pay-per-click auction model developed by Overture. Licensing their search results also was not profitable enough to pay for their scaling costs. They failed to develop a profitable business model, and sold out to Yahoo! for approximately $235 million, or $1.65 a share, in December of 2003.

*http://searchenginewatch.com/article/2066745/Inktomi-Spam-Database-Left-Open-To-Public

Copyright Ellis Horowitz, 2011-2022

- In April of 1997 Ask Jeeves was launched as a natural language search engine.
  - Ask Jeeves used human editors to try to match search queries.
  - Ask was powered by DirectHit for a while, which aimed to rank results based on their popularity, but that technology proved too easy to spam.
  - In 2000 the Teoma search engine was released, which uses clustering to organize sites by Subject Specific Popularity, which is another way of saying they tried to find local web communities. In 2001 Ask Jeeves bought Teoma to replace the DirectHit search technology.
  - On March 21, 2005 Barry Diller's IAC agreed to acquire Ask Jeeves for 1.85 billion dollars. IAC owns many popular websites like Match.com, Ticketmaster.com, and Citysearch.com, and is promoting Ask across their other properties.
  - In 2006 Ask Jeeves was renamed to Ask.

Copyright Ellis Horowitz, 2011-2022

- Google is a play on the word Googol, coined by Milton Sirotta; it refers to a 1 followed by 100 zeros, 10000000…..0
- A googol is bigger than the number of atoms in the universe
- Google was founded by Larry Page and Sergey Brin, two Stanford Univ. Computer Science graduate students
- In 1998 they built a prototype system called BackRub, dropped out of school, and tried to attract investors for their new company
- Google Inc. released a beta version on Sept. 7, 1998
- www.google.com was officially released on Sept. 21, 1999

Copyright Ellis Horowitz, 2011-2022

A Brief Chronology of Search Engines

Search Engine Basic Behavior

Copyright Ellis Horowitz 2013-2022

What is Web Search?

- **Providing access to heterogeneous, distributed information that is publicly available on the World Wide Web**
  - Information comes in many different formats
  - Most of the information has not been screened for accuracy
- **Multi-billion dollar business**
- **Source of new opportunities in marketing**
- **Strains the boundaries of trademark and intellectual property laws**
- **A source of unending technical challenges**

Copyright Ellis Horowitz, 2011-2022

Web Search Engine Definitions

- **"A search engine is a program designed to help find information stored on a computer system such as the World Wide Web, inside a corporate or proprietary network or a personal computer"** ***wikipedia***
  - ***search engine*** usually refers to a *Web* search engine, which searches for information on the public Web.
  - Other kinds of search engine are ***enterprise search engines***, which search on intranets,
  - ***personal search engines***, which search individual personal computers

Copyright Ellis Horowitz, 2011-2022

Basic Web Search Internals

Major Elements

1. The User
2. The Web
3. The Crawler/spider
4. The Indexer
5. The Ads
6. The Query Processor

Copyright Ellis Horowitz, 2011-2022

Web Search Engine Elements

- ***Spider*** **(a.k.a. crawler/robot) – builds corpus**
  - **Collects web pages recursively**
    - For each known URL, fetch the page, parse it, and extract new URLs
    - Repeat
  - **Additional pages come from direct submissions & other sources**
- **The *indexer* – creates inverted indexes**
  - Various policies wrt which words are indexed, capitalization, support for Unicode, stemming, support for phrases, etc.
- ***Query processor*** **– serves query results**
  - **Front end –** query reformulation, word stemming, capitalization, optimization of Booleans, etc.
  - **Back end –** finds matching documents and ranks them

Copyright Ellis Horowitz, 2011-2022

The Web

- No design/co-ordination
- Distributed content creation, linking
- Content includes truth, lies, obsolete information, contradictions …
- Data is stored in structured (databases), semi-structured (tables)…
- Scale larger than previous text corpora
- Growth – still expanding
- Content can be *dynamically generated*

Copyright Horowitz 2006-2022

The User

- **Diverse in background/training**
  - Users sometimes cannot tell the difference between a search bar from the URL address field (**Chrome conflates the two**)
  - Users rarely use the scroll bar, so key results must be at or near the top
- **Diverse in access methodology**
  - Increasingly, high bandwidth connectivity
  - Growing segment of mobile users: limitations of form factor – keyboard, display
- **Diverse in search methodology**
  - Search, search + browse,
  - Average query length ~ 2.5 terms
  - Has to do with what they're searching for
- **Poor comprehension of syntax**
  - Early engines offered rich syntax for queries – Boolean, phrase, etc.
  - Current engines hide these

Copyright Ellis Horowitz, 2011-2022

User's Information Needs Are Diverse

- Informational – want to learn about something (~40%)

  e.g. Low hemoglobin

- Navigational – want to go to that page (~25%)

  e.g. United Airlines

- Transactional – want to do something (web-mediated) (~35%)
  - Access a service — Los Angeles weather
  - Downloads — Mars surface images
  - Shop — Nikon CoolPix Camera
- Gray areas
  - Find a good hub — Car rental in Finland
  - Exploratory search "see what's there"

Copyright Ellis Horowitz, 2011-2022

Query Processing is Now Extremely Clever

- **Query processing involves much more than just matching query terms with document terms**
- **Semantic analysis of the query includes:**
  1. Determining the language of the query
  2. Filtering of unnecessary words from the query (stop words)
  3. Looking for specific types of queries, e.g.
     - **Personalities (triggered on names)**
     - **Cities (travel info, maps)**
     - **Medical info (triggered on names and/or results)**
     - **Stock quotes, news (triggered on stock symbol)**
     - **Company info …**
  4. Determining the user's location or the target location of the query
  5. Remembering previous queries
  6. Maintaining a user profile

Copyright Ellis Horowitz, 2011-2022

Google Maintains Your Recent Query History

Maintain previous queries
Helps to minimize typing
Allow users to remove old ones

Results are Holistic
A Person Query

File Edit View History Bookmarks Tools Help

george clooney - Google S...

https://www.google.com/#q=george+clooney

Google | george clooney | +Ellis | Share

Web | News | Images | Videos | Shopping | More | Search tools

About 37,700,000 results (0.23 seconds)

In the news

George Clooney, Amal Alamuddin Honeymoon in New British Home
Us Magazine - 4 hours ago
Newlyweds **George Clooney** and Amal Alamuddin have skipped the traditional far-flung ...

People: **George Clooney's** Wedding Cost About $1.6 Million
Yahoo! Voices - 2 days ago

George Clooney & Amal Alamuddin Could Nab His & Hers Nobel Peace Prizes, Friend Predicts
People Magazine - 23 hours ago

More news for **george clooney**

George Clooney - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**George_Clooney** - Wikipedia
George Timothy **Clooney** (born May 6, 1961) is an American actor and filmmaker. He has received three Golden Globe Awards for his work as an actor and two ...
Filmography - Talia Balsam - Amal Alamuddin - Nick Clooney

George Clooney & Amal Alamuddin Could Nab His & Hers ...
www.people.com/.../**george-clooney**-amal-alamuddin-nobel-peace-p... - People
23 hours ago - SEE 26 MORE **CLOONEY** WEDDING PHOTOS! Subscribe now to PEOPLE'S digital edition and get 26 BONUS photos from inside the ...

George Clooney - IMDb
www.imdb.com/name/nm0000123/ - Internet Movie Database

George Clooney

Actor

George Timothy Clooney is an American actor and filmmaker. He has received three Golden Globe Awards for his work as an actor and two Academy Awards, one for acting and the other for producing.
Wikipedia

Born: May 6, 1961 (age 53), Lexington, KY
Height: 5' 11" (1.80 m)
Spouse: Amal Alamuddin (m. 2014), Talia Balsam (m. 1989–1993)
Siblings: Adelia Clooney
Parents: Nina Bruce Warren, Nick Clooney

Includes the following:

- Latest news
- Biography
- Photos
- Basic facts
  - born
  - married
  - parents
  - career

Results are Holistic
A Place Query

las vegas - Google Search

https://www.google.com/search?q=las+vegas&oq=las+vegas&aqs=chrome..69i57j0l5.1920j0j8&source

Apps | CSCI 572 Home Page | CSCI 571 Home Page | CSCI 351 Home Page | Ellis Horowitz' Hom... | Computer Science D... | Other bookmarks

Google | las vegas | +Ellis | Share

Web | News | Images | Maps | Videos | More | Search tools

About 168,000,000 results (0.41 seconds)

Las Vegas - VEGAS.com™
Ad www.vegas.com/
Las Vegas Shows, Hotels and more. The Official **VEGAS** Travel Site™.
Vegas.com has 767 followers on Google+
Las Vegas Air + Hotel - Las Vegas Hotels - Headliners and Concerts

Attractions In Las Vegas - TravelNevada.com
Ad www.travelnevada.com/
Order Your Free Visitor's Guide & Plan Your Vacation To **Las Vegas**!
Travel Nevada has 1,238 followers on Google+

Las Vegas 5 Day Room Sale - SouthPointCasino.com
Ad www.southpointcasino.com/
South Point Hotel Casino Spa Rates From $35 Sun-Thurs $60 Fri. & Sat.
Restaurants - Hotel Packages & Promos - Nightlife & Entertainment - Casino

(Official City of Las Vegas Web Site)
www.lasvegasnevada.gov/ - Las Vegas
The City of **Las Vegas** (Official Government Site) ... City of **Las Vegas**: Serving You Online Rather Than In Line, Photo: Downtown **Las Vegas** ...

Las Vegas Hotels, Shows, Casinos, Restaurants, Maps and ...
www.lasvegas.com/
Your official What happens in Vegas, stays in Vegas resource. Plan hotels and things to do for your trip on the only official website of **Las Vegas**.
Shows & Events - Las Vegas Hotels - Air + Hotel Packages - Special Offers & Deals

Map: Las Vegas, Sunrise Manor, Spring Valley, Paradise, Henderson
Map data ©2014 Google

Las Vegas
26,005 followers on Google+
Follow

Las Vegas, officially the City of Las Vegas and often known as simply Vegas, is the most populous city in the U.S. state of Nevada and the county seat of Clark County. Wikipedia

Founded: May 15, 1905
Area: 135.9 sq miles (352 km²)
Weather: 87°F (31°C), Wind N at 5 mph (8 km/h), 5% Humidity
Local time: Thursday 9:32 AM

Includes the following:

Official site
Map
Essential facts
- founded
- area
- weather
- time
- population

Results are Holistic
An Hotel Query

sheraton times square hot

https://www.google.com/search?q=las+vegas&oq=las+vegas&aqs=chrome..69i57j0l5.1920j0j8&source

Apps | CSCI 572 Home Page | CSCI 571 Home Page | CSCI 351 Home Page | Ellis Horowitz' Hom... | Computer Science D... | Other bookmarks

Google | sheraton times square hotel nyc | +Ellis | Share

Web | Maps | Images | News | Shopping | More | Search tools

About 24,000,000 results (0.55 seconds)

Sheraton™ New York Hotel - Official Site - Our Best Rates
Ad www.sheraton.com/TimesSquare
Guaranteed. Book Now!
Sheraton Hotels & Resorts has 1,593 followers on Google+
811 7th Avenue, New York, NY
Photos | Features & Amenities
Make a Reservation | Special Offers

NYC Times Square Hotel - sofitel-new-york.com
Ad www.sofitel-new-york.com/
Sofitel **New York Hotel** in Manhattan near **Times Square**. Book Direct Now!
Make Your Reservation - Enjoy Business @ Sofitel - Sumptuous Terrace Suites

Sheraton New York Times Square Hotel: Hotel Near Time...
www.sheratonnewyork.com/
The **Sheraton New York Times Square Hotel** is located between Central Park and **Times Square**, and offers newly renovated spaces and sophisticated ...
3.2 ★★★☆☆ 141 Google reviews · Write a review - $239
811 7th Ave, New York, NY 10019
(212) 581-1000
Photos & Videos - Sheraton New York Times ...

Sheraton New York Times Square Hotel - Starwood Hotel...
www.starwoodhotels.com/sherato... Starwood Hotels & Resorts Worldwide
★★★½☆ Rating: 3.7 - 1,851 reviews

Map data ©2014 Google

Sheraton New York Times Square Hotel
$239 Book | Directions

Are you the business owner? | Feedback

Ads
Sheraton Times Square Hotel
sheratonnewyork.reservations.com/
Book **Sheraton Times Square**
For Limited Offers - Reserve Today!

Includes the following:
Main hotel website
Map
Address
Phone number
Price of a room
Directions

Google Retains a User's Entire Query History!

They claim that only I can see my history;
I have issued a total of 18,960 queries;

Graphs show my queries by hour and by week;

I can view my Web queries as distinct from my Image queries or my News queries, etc

As a result, Google now knows a great deal about us!

Search Engines are an Industry

- **The search engine industry is 20+ years old, having started with WebCrawler and Lycos in 1994 who sold banner ads as their business model**
- **Search engine revenue today**
  - **Google:** 2021:$257 Billion; 2020: $181 Billion; 2019: $162 Billion; 2018: $116 Billion; 2017: $109 Billion; 2016: $90 Billion; 2015: $74.5 Billion; 2014: $66 Billion; 2013: $37 Billion
  - **Baidu:** 2021: $31 Billion; 2020: $16.4 Billion; 2019: $15 Billion; : $11.3 Billion; 2017: $13 Billion; 2016: $10.1 Billion; 2015: $10.2 Billion; 2014: 8.0 Billion
  - **Yahoo:** 2021: 5.2Billion; 2019: 6.97Billion; 2018: 3.03 Billion; 2017: 3.0 Billion; 2016: 2.98 Billion;2015: $4.9 Billion; 2014: 4.6 Billion; 2013: 4.6Billion
  - **Bing:** 2020 $7.74 Billion; 2019: $7.63 Billion; 2018: $7.01 Billion
    - Microsoft says that in Q1 2016 Bing became profitable

Copyright Ellis Horowitz, 2011-2022

Google is a Monopoly Gatekeeper for the Internet

- The US is suing Google for anti-Trust violations
  - Google claims it has strong competition in search!!!
  - Google has sweetheart deals with Apple andpays Apple $8+ Billion/year to be their default search engine
- Google maintains the largest index of the web
  - Some websites actually deny access to crawlers other than Google and Bing as these other crawlers bring in little traffic and consume server cycles
  - Only Google and Bing have the resources to maintain such a large index, e.g. DuckDuckGo no longer crawls the web and uses Bing's index

Copyright Ellis Horowitz, 2011-2022

Crawlers and Crawling

There are Many Crawlers

- **A web crawler is a computer program that visits web pages in an organized way**
  - Sometimes called a spider or robot
- **A list of web crawlers can be found at**

  http://en.wikipedia.org/wiki/Web_crawler

  Google's crawler is called googlebot, see

http://support.google.com/webmasters/bin/answer.py?hl=en&answer=182072

- **Yahoo's web crawler is/was called Yahoo! Slurp, see**

http://en.wikipedia.org/wiki/Yahoo!_Search

- **Bing uses five crawlers**
  - Bingbot, standard crawler
  - Adidxbot, used by Bing Ads
  - MSNbot, remnant from MSN, but still in use
  - MSNBotMedia, crawls images and video
  - BingPreview, generates page snapshots
- **For details see**: http://www.bing.com/webmaster/help/which-crawlers-does-bing-use-8c184ec0

Copyright Ellis Horowitz, 2011-2022

Web Crawling Issues

- **How to crawl?**
  - *Quality*: how to find the "Best" pages first
  - *Efficiency*: how to avoid duplication (or near duplication)
  - *Etiquette*: behave politely by not disturbing a website's performance
- **How much to crawl? How much to index?**
  - *Coverage*: What percentage of the web should be covered?
  - *Relative Coverage:* How much do competitors have?
- **How often to crawl?**
  - *Freshness*: How much has changed?
  - How much has <u>really</u> changed?

Copyright Ellis Horowitz, 2011-2022

Simplest Crawler Operation

- **Initialize (begin with known "seed" pages)**
- **Loop: Fetch and parse a page**
  - Place the page in a database
  - Extract the URLs within the page
  - Place the extracted URLs on a queue
  - Fetch a URL on the queue and repeat

Copyright Ellis Horowitz, 2011-2022

Crawling Picture

20 *Web crawling and indexes*

20.1 Overview

Web crawling is the process by which we gather pages from the Web, in order to index them and support a search engine. The objective of crawling is to quickly and efficiently gather as many useful web pages as possible, together with the link structure that interconnects them. In Chapter 19 we studied the complexities of the Web stemming from its creation by millions of uncoordinated individuals. In this chapter we study the resulting difficulties for crawling the Web. The focus of this chapter is the component shown in Figure 19.7 as *web crawler*; it is sometimes referred to as a *spider*.

WEB CRAWLER
SPIDER

The goal of this chapter is not to describe how to build the crawler for a full-scale commercial web search engine. We focus instead on a range of issues that are generic to crawling from the student project scale to substantial research projects. We begin (Section 20.1.1) by listing desiderata for web crawlers, and then discuss in Section 20.2 how each of these issues is addressed. The remainder of this chapter describes the architecture and some implementation details for a distributed web crawler that satisfies these features. Section 20.3 discusses distributing indexes across many machines for a web-scale implementation.

20.1.1 Features a crawler *must* provide

We list the desiderata for web crawlers in two categories: features that web crawlers *must* provide, followed by features they *should* provide.

Robustness: The Web contains servers that create *spider traps*, which are generators of web pages that mislead crawlers into getting stuck fetching an infinite number of pages in a particular domain. Crawlers must be designed to be resilient to such traps. Not all such traps are malicious; some are the inadvertent side-effect of faulty website development.

Online edition (c) 2009 Cambridge UP

Our textbook

Simple Picture – Complications

- **Crawling the entire web isn't feasible with one machine**
  - But all of the above steps can be distributed
- **Challenges**
  - **Handling/Avoiding malicious pages**
    - Some pages contain spam
    - Some pages contain spider traps – especially dynamically generated pages
  - **Even non-malicious pages pose challenges**
    - Latency/bandwidth to remote servers can vary widely
    - Robots.txt stipulations can prevent web pages from being visited
    - How can one avoid mirrored sites and duplicate pages
  - **Maintain politeness – don't hit a server too often**

Copyright Ellis Horowitz, 2011-2022

Robots.txt

- **There is a protocol that defines the limitations for a web crawler as it visits a website; its definition is here**
  - http://www.robotstxt.org/orig.html
- **The website announces its request on what can(not) be crawled by placing a robots.txt file in the root directory**
  - e.g. see

  http://www.ticketmaster.com/robots.txt

Copyright Ellis Horowitz, 2011-2022

Robots.txt Example

- **No robot visiting this domain should visit any URL starting with "/yoursite/temp/":**

```
User-agent: *
Disallow: /yoursite/temp/
```

- **Directives are case sensitive**
- **Additional symbols allowed in the robots.txt directives include:**
  - '*‘ - matches a sequence of characters
  - '$' - anchors at the end of the URL string
- **Example of '*':**

```
User-agent: Slurp
Allow: /public*/
Disallow: /*_print*.html
Disallow: /*?sessionid
```

Copyright Ellis Horowitz, 2011-2022

Websites Give Preference to Googlebot

- In researching how websites treat crawlers, Zack Maril looked at 17 million robots.txt files and discovered many places where Google is given an advantage; e.g. see
- https://www.sciencedirect.com/robots.txt
- E.g.

```
# go away ? tell all others not in the list below to stay out!
User-agent: *
Disallow: /
# As of 02/09/2021, there are 10 crawlers welcomed by SD
User-agent: Googlebot
Disallow: /cache/MiamiImageURL/
Disallow: /science?_ob=MiamiCaptionURL*
```

Copyright Ellis Horowitz, 2011-2022

A Study of Robots.txt Files

- ***Determining bias to search engines from robots.txt*, Giles, Sun, Zhuang, Penn State**

Favored Robots (Sample size $N = 2925$)				
robot name	N_{favor}	N_{disf}	$\Delta P(r)$	σ
google	877	25	0.2913	0.0084
yahoo	631	34	0.2041	0.0075
msn	349	9	0.1162	0.0059
scooter	341	15	0.1104	0.0058
lycos	91	5	0.0294	0.0031
netmechanic	84	10	0.0253	0.0029
htdig	15	3	0.0041	0.0012
teoma	13	3	0.0034	0.0011
oodlebot*	8	0	0.0027	0.0010
momspider	6	0	0.0021	0.0008
Disfavored Robots (Sample size $N = 2925$)				
robot name	N_{favor}	N_{disf}	$\Delta P(r)$	σ
msiecrawler	0	85	-0.0291	0.0031
ia_archiver	7	55	-0.0164	0.0023
cherrypicker	0	37	-0.0126	0.0021
emailsiphon	3	34	-0.0106	0.0019
roverbot	2	27	-0.0085	0.0017
psbot	0	23	-0.0079	0.0016
webzip	0	21	-0.0072	0.0016
wget	1	22	-0.0072	0.0016
linkwalker	2	20	-0.0062	0.0015
asterias	0	18	-0.0062	0.0015

Table 2. Top 10 favored and disfavored robots. σ is the standard deviation of $\Delta P(r)$.

Figure 2. Most frequently used robot names in robots.txt files. The height of the bar represents the number of times a robot appeared in our dataset.

Basic Search Strategies

Breadth-first Search

Examine all pages at level i before examining pages at level i+1

Basic Search Strategies (cont)

Depth-first Search

At each step move to a page down the tree

Web Wide Crawl (328M pages) [Najo01]

Page Rank is an algorithm developed by Google for determining the value of a page

Crawling Algorithm – Version 2

Initialize queue (Q) with initial set of known URL's.
Loop until Q empty or page or time limit exhausted:
    Pop a URL, call it L, from the front of Q.
    If L is not an HTML page (e.g. .gif, .jpeg, ….)
        continue the loop
    If L has already been visited, continue the loop.
    Download page, P, for L
    If cannot download P (e.g. 404 error, robot excluded)
        continue loop
    Index P (e.g. add to inverted index and store cached copy)
    Parse P to obtain list of new links N.
    Append N to the end of Q
End loop

Queueing Strategy

- **How new links are added to the queue determines the search strategy.**
- **FIFO (append to end of Q) gives breadth-first search.**
- **LIFO (add to front of Q) gives depth-first search.**
- **Heuristically ordering the Q gives a "focused crawler" that directs its search towards "interesting" pages; e.g.**
  - A document that changes frequently could be moved forward
  - A document whose content appears relevant to some topic can be moved forward
  - e.g. see Focused Crawling: A New Approach by S. Chakrabarti et al
  - https://www.sciencedirect.com/science/article/pii/S1389128699000523
- **One way to re-order the URLs on the queue is to:**
  - Move forward URLs whose In-degree is large
  - Move forward URLs whose PageRank is large
    - We will discuss the PageRank algorithm later

Copyright Ellis Horowitz, 2011-2022

Avoiding Page Duplication

- **A crawler must detect when revisiting a page that has already been crawled (Remember: the web is a graph not a tree).**
- **Therefore, a crawler must efficiently index URLs as well as already visited pages**
- **To determine if a URL has already been seen,**
  - Must store URLs in a standard format (discussed ahead)
  - Must develop a fast way to check if a URL has already been seen
- **To determine if a new page has already been seen,**
  - Must develop a fast way to determine if an *identical* page was already indexed
  - Must develop a fast way to determine if a *near-identical* page was already indexed

Copyright Ellis Horowitz, 2011-2022

Link Extraction

- **Must find all links in a page and extract URLs;**

```
var links = document.querySelectorAll("a");
for (var i = 0; i < links.length; i++) {
    var link = links[i].getAttribute("href");
    console.log(link); }
```

  – But URLs occur in tags other than <a>, e.g.
  – <frame src="site-index.html">, <area, href="…">, <meta>, <link>, <script>
- **Relative URL's must be completed, e.g. using current page URL or <base> tag**
  – <a href="proj.html"> to http://www.myco.com/special/tools/proj.html
  – <a href="../outline/syllabus.html"> to http://www.myco.com/special/outline/syllabus.html
- **Two Anomalies**
  1. Some anchors don't have links, e.g. <a name="banner">
  2. Some anchors produce dynamic pages which can lead to looping
     <a href=http://www.mysite.com/search?x=arg1&y=arg2>

Copyright Ellis Horowitz, 2011-2022

Representing URLs

- **URLs are rather long, 80 bytes on the average, implying 1 trillion URLs will require 80 Terabytes**
    - Recently Google reported finding 30 trillion unique URLs, which by the above would require 2400 terabytes (or 2.4 petabytes) to store

1. **One Proposed Method: To determine if a new URL has already been seen**
   - First hash on host/domain name, then
   - Use a trie data structure to determine if the path/resource is the same as one in the URL database
2. **Another Proposed Method: URLs are sorted lexicographically and then stored as a delta-encoded text file**
    - Each entry is stored as the difference (delta) between the current and previous URL; this substantially reduces storage
    - However, restoring the actual URL is slower, requiring all deltas to be applied to the initial URL
    - To improve speed, checkpointing (storing the full URL) is done periodically

Copyright Ellis Horowitz, 2011-2022

Trie for URL Exact Matching

- **Simplest (and worst) algorithm to determine if a new URL is in your set**
  - `grep -i <search_url> <url_file>`
  - **For *N* URLs and maximum length *K*, time is *O(NK)***
- **Characteristics of tries**
  - **They share the same prefix among multiple "words"**
  - **Each path from the root to a leaf corresponds to one "word"**
  - ***Endmarker* symbol, $, at the ends of all words**
    - **To avoid confusion between words with almost identical elements**
      - **Assume all words are $ terminated**

If we store *N* URLs, each of maximum length *K*, in a binary search tree, then the search time is *O(K*log N); h*owever, using a trie, the search time is *O(K)*, at the expense of more storage

Copyright Ellis Horowitz, 2011-2022

Why Normalizing URLs is Important

- For example, all the following URLs have the same meaning (return the same web page), but different hashes:
  - http://www.google.com
  - http://www.google.com/
  - https://www.google.com
  - www.google.com
  - google.com
  - google.com/
  - google.com.

Copyright Ellis Horowitz, 2011-2022

Normalizing URLs (4 rules)

1. **Convert the scheme and host to lower case. The scheme and host components of the URL are case-insensitive.**
   - HTTP://www.Example.com/ → http://www.example.com/
2. **Capitalize letters in escape sequences. All letters within a percent-encoding triplet (e.g., "%3A") are case-insensitive, and should be capitalized. Example:**
   - http://www.example.com/a%c2%b1b → http://www.example.com/a%C2%B1b
3. **Decode percent-encoded octets of unreserved characters.**
   http://www.example.com/%7Eusername/ → http://www.example.com/~username/
4. **Remove the default port. The default port (port 80 for the "http" scheme) may be removed from (or added to) a URL. Example:**
   - http://www.example.com:80/bar.html →

   http://www.example.com/bar.html

- **See https://en.wikipedia.org/wiki/URL_normalization**

Copyright Ellis Horowitz, 2011-2022

Avoiding Spider Traps

- **A spider trap is when a crawler re-visits the same page over and over again**
- **The most well-known spider trap is the one created by the use of Session ID's**
  - J2EE, ASP, .NET, and PHP all provide session ID management
- **A Session ID is often used to keep track of visitors, and some sites puts a unique ID in the URL:**
  - An example is www.webmasterworld.com/page.php?id=264684413484654 (**Note** this URL doesn't exist).
    Each user gets a unique ID and it's often requested from each page.
    The problem here is when Googlebot comes to the page, it spiders the page and then leaves, it goes to another page and it finds a link to the previous page, but since it has been given a different session id now, the link shows up as another URL.

- One way to avoid such traps is for the crawler to be careful when the querystring "ID=" is present in the URL
- Another technique is to monitor the length of the URL, and stop if the length gets "too long"

Copyright Ellis Horowitz, 2011-2022

Handling Spam Web Pages

- The **first generation** of spam web pages consisted of pages with a high number of repeated terms, so as to score high on search engines that ranked by word frequency
  - Words were typically rendered in the same color as the background, so as to not be visible, but still count
- The **second generation** of spam used a technique called *cloaking*;
  - When the web server detects a request from a crawler, it returns a different page than the page it returns from a user request
  - The page is mistakenly indexed

Google's example of keyword stuffing:

> *We sell custom cigar humidors. Our custom cigar humidors are handmade. If you're thinking of buying a custom cigar humidor, please contact our custom cigar humidor specialists at custom.cigar.humidor@example.com.*

- A third generation, called a doorway page, contains text and metadata chosen to rank highly on certain search keywords, but when a browser requests the doorway page it instead gets a more "commercially oriented" (more ads) page
- *Cloaking* and *doorway pages* are not permitted according to Google's webmaster suggestions

  See http://support.google.com/webmasters/bin/answer.py?hl=en&answer=66355

The Mercator Web Crawler

The diagram points out all of the key elements of a crawler;

Notice

1. The DNS caching server
2. Use of UDP for DNS
3. Load and thread monitor
4. Parallel threads waiting for a page to download

Copyright Ellis Horowitz, 2011-2022

Measuring and Tuning a Crawler

- **Measuring and tuning a crawler for peak performance eventually reduces to**
  - Improving URL parsing speed
  - Improving network bandwidth speed
  - Improving fault tolerance
- **More Issues (some of which are discussed ahead)**
  - Refresh Strategies: how often is the process re-started
  - Detecting duplicate pages
  - Detecting mirror sites
  - Speeding up DNS lookup (see previous slide)
  - URL normalization (discussed earlier)
  - Handling malformed HTML

Copyright Ellis Horowitz, 2011-2022

DNS caching, pre-fetching and resolution

– ***A common operating system's implementation of DNS lookup is blocking: only one outstanding request at a time*; so**

1. **DNS caching: build a caching server that retains IP-domain name mappings previously discovered**
2. **Pre-fetching client**
   - once a page is parsed,
     - immediately make DNS resolution requests to the caching server; and
     - if unresolved, use UDP (User Datagram Protocol) to resolve from the DNS server
3. **Customize the crawler so it allows issuing of many resolution requests simultaneously; there should be many DNS resolvers**

Copyright Ellis Horowitz, 2011-2022

Multi-Threaded Crawling

- **One bottleneck is network delay in downloading individual pages.**
- **It is best to have multiple threads running in parallel each requesting a page from a different host.**
  - a **thread** of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler.
  - In most cases, a thread is a component of a process.
  - Multiple threads can exist within the same process and share resources
- **Distribute URL's to threads to guarantee equitable distribution of requests across different hosts to maximize through-put and avoid overloading any single server.**
- **Early Google spider had multiple coordinated crawlers with about 300 threads each,**
  - together they were able to download over 100 pages per second back in 2010
  - It is estimated that in 2021 Google downloads ~50,000 pages/seconds or 4billion+ in a day, see https://www.quora.com/How-many-pages-are-Google-bots-crawling-every-second

Copyright Ellis Horowitz, 2011-2022

Distributed Crawling Approaches

- **Once the crawler program itself has been optimized, the next issue to decide is how many crawlers will be running at any time**
- **Scenario 1:** A *centralized crawler* controling a set of parallel crawlers all running on a LAN
  - A *parallel crawler* consists of multiple crawling processes communicating via local network (sometimes called an intra-site parallel crawler)
- **Scenario 2:** A *distributed set of crawlers* running on widely distributed machines, with or without cross communication

Copyright Ellis Horowitz, 2011-2022

Distributed Model

- **If crawlers are running in diverse geographic locations, how do we organize them**
  - By country, by region, by available bandwidth
  - Distributed crawlers must periodically update a master index
  - But incremental update is generally "cheap"
    - Why? Because
      - a. you can compress the update, and
      - b. you need only send a differential update

      both of which will limit the required communication

Copyright Ellis Horowitz, 2011-2022

Issues and Benefits of Distributed Crawling

- **Benefits:**
  - scalability: for large-scale web-crawls
  - costs: use of cheaper machines
  - network-load dispersion and reduction: by dividing the web into regions and crawling only the nearest pages
- **Issues:**
  - overlap: minimization of multiple downloaded pages
  - quality: depends on the crawl strategy
  - communication bandwidth: minimization

Copyright Ellis Horowitz, 2011-2022

Coordination of Distributed Crawling

– **Three strategies**

1. **Independent:**
   - no coordination, every process follows its extracted links
2. **Dynamic assignment:**
   - a central coordinator dynamically divides the web into small partitions and assigns each partition to a process
3. **Static assignment:**
   - Web is partitioned and assigned without a central coordinator before the crawl starts

Copyright Ellis Horowitz, 2011-2022

Static Assignment

Links from one partition to another (inter-partition links) can be handled in one of three ways:

1. *Firewall mode*:
   a process does not follow any inter-partition link
2. *Cross-over mode*:
   a process also follows inter-partition links and possibly discovers also more pages in its partition
3. *Exchange mode*:
   processes exchange inter-partition URLs; this mode requires communication

Copyright Ellis Horowitz, 2011-2022

Classification of Parallel Crawlers

- **If exchange mode is used, communication can be limited by:**
  - Batch communication: every process collects some URLs and sends them in a batch
  - Replication: the *k* most popular URLs are replicated at each process and are not exchanged (previous crawl or on the fly)
- **Some ways to partition the Web:**
  - URL-hash based: this yields many inter-partition links
  - Site-hash based: reduces the inter partition links
  - Hierarchical: by TLD, e.g. .com domain, .net domain …
- **General Conclusions of Cho and Garcia-Molina**
  - Firewall crawlers attain good, general coverage with low cost
  - Cross-over ensures 100% quality, but suffer from overlap
  - Replicating URLs and batch communication can reduce overhead

Copyright Ellis Horowitz, 2011-2022

Freshness

- **The behavior of a Web crawler is the outcome of a combination of policies:**
  - A *selection policy* that states which pages to download.
  - A *re-visit policy* that states when to check for changes to the pages.
  - A *politeness policy* that states how to avoid overloading websites.
  - A *parallelization policy* that states how to coordinate distributed web crawlers.

Copyright Ellis Horowitz, 2011-2022

Keeping Spidered Pages Up to Date

- **Web is very dynamic: many new pages, updated pages, deleted pages, etc.**
- **Periodically check crawled pages for updates and deletions:**
  - Just look at LastModified indicator to determine if page has changed, only reload entire page if needed
- **Track how often each page is updated and preferentially return to pages which are historically more dynamic.**
- **Preferentially update pages that are accessed more often to optimize freshness of more popular pages.**

Copyright Ellis Horowitz, 2011-2022

Implications for a Web Crawler

- **A *steady crawler* runs continuously without pause**
  - Typically search engines use multiple crawlers
- **When a crawler replaces an old version by a new page, does it do it "in-place" or "shadowing"**
  - Shadowing implies a new set of pages are collected and stored separately and all are updated at the same time
  - The above implies that queries need to check two databases, the current database and the database of new pages
  - Shadowing either slows down query processing or decreases freshness
- **Conclusions:**
  - running multiple types of crawlers is best
  - Updating in-place keeps the index current

Copyright Ellis Horowitz, 2011-2022

Cho and Garcia-Molina, 2000

- **Two simple re-visiting policies**
  - **Uniform policy:** This involves re-visiting all pages in the collection with the same frequency, regardless of their rates of change.
  - **Proportional policy:** This involves re-visiting more often the pages that change more frequently. The visiting frequency is directly proportional to the (estimated) change frequency.
- **Cho and Garcia-Molina proved the surprising result that, in terms of average freshness, the uniform policy outperforms the proportional policy in both a simulated Web and a real Web crawl.**
- **The explanation for this result comes from the fact that, when a page changes too often, the crawler will waste time by trying to re-crawl it too fast and still will not be able to keep its copy of the page fresh.**
- **To improve freshness, we should penalize the elements that change too often**

Copyright Ellis Horowitz, 2011-2022

Help the Search Engine Crawler Creating a SiteMap

- A sitemap is a list of pages of a web site accessible to crawlers
- This helps search engine crawlers find pages on the site
- XML is used as the standard for representing sitemaps
- Here is an example of an XML sitemap for a three page website

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
   <loc>http://www.example.com/?id=who</loc>
   <lastmod>2009-09-22</lastmod>
   <changefreq>monthly</changefreq>
   <priority>0.8</priority> </url>
<url>
   <loc>http://www.example.com/?id=what</loc>
   <lastmod>2009-09-22</lastmod>
   <changefreq>monthly</changefreq>
   <priority>0.5</priority> </url>
<url>
<loc>http://www.example.com/?id=how</loc>
<lastmod>2009-09-22</lastmod>
<changefreq>monthly</changefreq>
<priority>0.5</priority> </url>
</urlset>
```

Back in 2006 Google introduced the sitemap format; now Bing, Yahoo, and Ask also support sitemaps

See the Google, Bing, Yahoo, Ask announcement: http://www.google.com/press/pressrel/sitemapsorg.html

Copyright Ellis Horowitz, 2011-2022

General Sitemap Guidelines

- Use consistent, fully-qualified URLs. Google will crawl your URLs exactly as listed
- A sitemap can be posted anywhere on your site, but a sitemap affects only descendants of the parent directory
- Don't include session IDs from URLs in your sitemap.
- Sitemap files must be UTF-8 encoded, and URLs escaped appropriately.
- If you have two versions of a page, list in the sitemap only the one you prefer to appear in search results
- If you have alternate pages for different languages or regions, you can use hreflang to indicate the alternate URLs.
-
```
<head>
 <title>Widgets, Inc</title>
  <link rel="alternate" hreflang="en-gb"
       href="https://en-gb.example.com/page.html" />
  <link rel="alternate" hreflang="en-us"
       href="https://en-us.example.com/page.html" />
  <link rel="alternate" hreflang="en”
      href="https://en.example.com/page.html" />
```

- There are many sitemap generator tools, e.g. https://slickplan.com/sitemap

Copyright Ellis Horowitz, 2011-2022

Google Crawlers

- Google now uses multiple crawlers
  - APIs-Google
  - AdSense
  - AdsBot Mobile Web Android
  - AdsBot Mobile Web
  - AdsBot
  - Googlebot Images
  - Googlebot News
  - Googlebot Video
  - Googlebot (desktop)
  - Googlebot (smartphone)
  - Mobile AdSense
  - Mobile Apps Android
  - Feedfetcher
  - Google Read Aloud

Crawler	User agent token (product token)	Full user agent string
APIs-Google	APIs-Google	APIs-Google (+https://developers.google.com/webmasters/APIs-Google.html)
AdSense	Mediapartners-Google	Mediapartners-Google
AdsBot Mobile Web Android (Checks Android web page ad quality)	AdsBot-Google-Mobile	Mozilla/5.0 (Linux; Android 5.0; SM-G920A) AppleW (KHTML, like Gecko) Chrome Mobile Safari (compati AdsBot-Google-Mobile; +http://www.google.com/mobile/adsbot.html)
AdsBot Mobile Web (Checks iPhone web page ad quality)	AdsBot-Google-Mobile	Mozilla/5.0 (iPhone; CPU iPhone OS 9_1 like Mac O AppleWebKit/601.1.46 (KHTML, like Gecko) Version/ Mobile/13B143 Safari/601.1 (compatible; AdsBot-Go Mobile; +http://www.google.com/mobile/adsbot.html
AdsBot (Checks desktop web page ad quality)	AdsBot-Google	AdsBot-Google (+http://www.google.com/adsbot.html
Googlebot Images	• Googlebot-Image • Googlebot	Googlebot-Image/1.0
Googlebot News	• Googlebot-News • Googlebot	Googlebot-News
Googlebot Video	• Googlebot-Video • Googlebot	Googlebot-Video/1.0
Googlebot (Desktop)	Googlebot	• Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)

For details see
https://support.google.com/webmasters/answer/1061943?hl=en
see also Google's tool for checking how Googlebot sees your website
https://support.google.com/webmasters/answer/6066468?rd=2

Copyright Ellis Horowitz, 2011-2022

Google's Googlebot

- **Begins with a list of webpage URLs generated from previous crawls**
- **Uses Sitemap data provided by webmasters**
- **Many versions of Googlebot are run on multiple machines located near the site they are indexing**
- **Googlebot cannot see within Flash files, audio/video tracks, and content within programs**
- **Advice**
  - To prevent "File not found" in a website's error log, create an empty robots.txt file
  - To prevent Googlebot from following any links on a page, use "nofollow" meta tag
  - To prevent Googlebot from following an individual link, add "rel='nofollow'" attribute to the link

Copyright Ellis Horowitz, 2011-2022

To verify That The Visitor is Googlebot

- Method 1
  1. Run a reverse DNS lookup on the accessing IP address from your logs, using the host command.
  2. Verify that the domain name is either googlebot.com or google.com.
  3. Run a forward DNS lookup on the domain name retrieved in step 1 using the host command on the retrieved domain name.
  4. Verify that it's the same as the original accessing IP address from your logs.
- Method 2
  - match the crawler's IP address to the list of Googlebot IP addresses


```
{
  "creationTime": "2022-07-21T15:54:44.265580",
  "prefixes": [
        {"ipv6Prefix": "2001:4860:4801:10::/64"},
        {"ipv6Prefix": "2001:4860:4801:11::/64"},
        {"ipv6Prefix": "2001:4860:4801:12::/64"},
        {"ipv6Prefix": "2001:4860:4801:13::/64"},
        {"ipv6Prefix": "2001:4860:4801:14::/64"},
        {"ipv6Prefix": "2001:4860:4801:15::/64"},
        {"ipv6Prefix": "2001:4860:4801:16::/64"},
        {"ipv6Prefix": "2001:4860:4801:17::/64"},
        {"ipv6Prefix": "2001:4860:4801:18::/64"},
        {"ipv6Prefix": "2001:4860:4801:19::/64"},
        {"ipv6Prefix": "2001:4860:4801:1a::/64"},
        {"ipv6Prefix": "2001:4860:4801:1b::/64"},
```


Copyright Ellis Horowitz, 2011-2022

Controlling How Often Googlebot Visits Your Site

- The term *crawl rate* means how many requests per second Googlebot makes to your site when it is crawling it: for example, 5 requests per second.
- You cannot change how often Google crawls your site, but if you want Google to crawl new or updated content on your site, you can request a recrawl; for details see

  https://developers.google.com/search/docs/advanced/crawling/ask-google-to-recrawl
- You can reduce the crawl rate by
  - returning pages with 500, 503 or 429 http status codes
  - Setting a new rate in the Search Console
  - A video discussing Google's Crawl Status Report can be found here: https://support.google.com/webmasters/answer/9679690

Copyright Ellis Horowitz, 2011-2022

Googlebot is Really Chrome Re-packaged

- **Why:** browsers don't just render the DOM hierarchy of HTML, they include transformations via CSS and JavaScript, and for Googlebot to extract the most meaningful features from a web page it would be necessary to have access to these transformations
  - Therefore Googlebot must understand and execute JavaScript code
- **Conclusion:** Googlebot and Chrome share a great deal of code
- **Googlebot processes web pages with JavaScript in 3 phases**
  1. Crawling – processing all links
  2. Rendering – executing JS and then looping back to 1
  3. Indexing
- As of 2019 Googlebot runs the latest Chromium rendering engine
- **Note: Server-side rendering saves Googlebot from rendering the page**

De-Duplication

The Definition of De-Duplication

- *De-Duplication* – the process of identifying and avoiding essentially identical web pages
- The term is often used in connection with *locker storage* where only a single copy of a file is stored and multiple links to the single file are managed
  - Whether this strategy effectively saves space is not clear and needs analysis for each particular application
  - However, this is ***not*** the meaning of the term that we are concerned about in this class
- **With respect to *web crawling*, de-duplication essentially refers to the identification of identical and nearly identical web pages and indexing only a single version to return as a search result**

Copyright 2011-2022 Ellis Horowitz

What are Web Duplicates?

- **One example is the same page, referenced by different URLs**

http://espn.go.com

http://www.espn.com

- **How can two URLs differ yet still point to the same page?**
  - **the URL's host name can be distinct (virtual hosts) sharing the same document folder,**
  - **the URL's protocol can be distinct (http, https), but still deliver the same document**
  - **the URL's path and/or page name can be distinct**

Copyright 2011-2022 Ellis Horowitz

Totally Distinct URLs May Deliver the Same Page

- At one time* all 3 URLs below pointed to the identical page
- Structural Classification of Proteins
  - http://scop.mrc-lmb.cam.ac.uk/scop
  - http://scop.berkeley.edu/
  - http://scop.protres.ru/
  - **The three URLs have distinct domain names, but all redirect to the same page**

*** At least they did when I took this snapshot, no longer**

Copyright 2011-2022 Ellis Horowitz

Near Duplicates-An Example

Another example is two web pages whose content differs slightly

Two copies of www.nytimes.com snapshot within a few seconds of each other;
The pages are essentially identical except for the ads at the top and the photo in the middle

Spotting Near Duplicates

- In examining a web page a search engine may want to ignore ads, navigation links and other elements that do not specifically relate to the contents of the web page
- One way to do this is to delve into the structure of a web page and focus on content blocks
- E.g. the Document Object Model for HTML displays a web page as a tree hierarchy
  - Document
  - Head
  - Body
- However this is time consuming

Web Duplicates-Mirroring

- Mirroring is the systematic replication of web pages across hosts.
  - Mirroring is the **single largest cause** of duplication on the web
- Host1/α and Host2/β are mirrors iff

  For all (or most) paths p such that when

  http://Host1/ α / p exists

  http://Host2/ β / p exists as well

  with identical (or near identical) content, and vice versa.

Copyright 2011-2022 Ellis Horowitz

List of Apache Mirror Sites

List of countries

281 sites in 55 regions

the status of apache mirrors - Mozilla Firefox

http://www.apache.org/mirrors/

the status of apache mirrors

date : Mon Jan 22 00:19:01 2007 (GMT)
last check : Mon Jan 22 00:19:01 2007 (GMT)

The Apache Software Foundation
http://www.apache.org/

- How do I become an Apache mirror site?

regions

BACKUP ar at au ba be bg br bs ca ch cl cn co cr cz de dk ee es
fr ge gr hk hr hu id ie il is it jp kr li lu lv mx my nl no pl pt ro
ru se sg si sk th tr tw ua uk us za

report

281 sites in 55 regions				
0 bad -- 73 older than 2.2 days -- 6 unreachable for more than 8 hours				
last probes : 275 were ok, 3 had no time, 3 had site not found				
apache site -- home	type	mirror age, daily stats	last probe, probe stats	last stat
BACKUP				
www.eu.apache.org @	http	4 hours	3 hours	ok
www.apache.org @	http	4 hours	4 hours	ok
argentina				
apache.mesi.com.ar @	http	25.2 days	1 hour	ok
apache.xmundo.com.ar @	http	82.5 days	5 hours	ok
apache.localhost.net.ar @	http	6 hours	3 hours	ok

Done 1.125s

Copyright 2011-2022 Ellis Horowitz

Two Sample Mirror Sites

Site in Australia

Note identical directories

Site in Argentina

Copyright 2011-2022 Ellis Horowitz

Why Detect Exact Duplicates

- ***Smarter crawling***
  - Avoid returning many duplicate results to a query
  - Allow fetching from the fastest or freshest server
- ***Better connectivity analysis***
  - By combining in-links from the multiple mirror sites to get an accurate PageRank (measure of importance)
  - Avoid double counting out-links
- ***Add redundancy in result listings***
  - "If that fails you can try: <mirror>/samepath"
- ***Reduce Crawl Time***: Crawlers need not crawl pages that are identical or near identical
- ***Ideally***: given the web's scale and complexity, priority must be given to content that has **not** already been seen before or has recently changed
  - Saves resources (on the crawler end, as well as the remote host)
  - Increases crawler politeness
  - Reduces the analysis that a crawler will have to do later

Copyright 2011-2022 Ellis Horowitz

Why Detect Near Duplicates

- **Clustering**
  - Given a news article some people might wish to see "related articles" describing the same event
- **Data extraction**
  - Given a collection of similar pages, e.g. movie reviews, a search engine can extract and categorize the information
- **Plagiarism**
  - Identify pairs that seem to have significantly borrowed from each other
- **Spam detection**
  - Spammers typically send similar emails en masse, so one can use near-similarity techniques to identify the spam
- **Duplicates within a domain**
  - To identify near-duplicates arising out of revisions, modifications, copying or merging of documents

Copyright 2011-2022 Ellis Horowitz

Solving the Duplicate/Near-Duplicate Detection Problems

1. ***Duplicate Problem*: Exact match;**
   - Solution: compute fingerprints using cryptographic hashing
   - Useful for URL matching and also works for detecting identical web pages
   - Hashes can be stored in sorted order for *log N* access
2. ***Near-Duplicate Problem*: Approximate match**
   - Solution: compute the syntactic similarity with an edit-distance measure, and
   - Use a similarity threshold to detect near-duplicates
     - e.g., Similarity > 80% => Documents are "near duplicates"

- The remaining slides are devoted to specific methods for duplicate and near duplicate detection

Copyright 2011-2022 Ellis Horowitz

Using a Cryptographic Hash Function to Convert a Web Page to a Number

- A **cryptographic hash function** is a hash function which takes an input (or 'message') and returns a fixed-size alphanumeric string, which is called the **hash value** (sometimes called a **message digest**, **digital fingerprint**, **digest** or a **checksum**).
- The cryptographic hash function has four main properties:
  1. It is extremely easy (i.e. fast) to calculate a hash for any given data.
  2. It is extremely computationally difficult to calculate an alphanumeric text that has a given hash.
  3. A small change to the text yields a totally different hash value.
  4. It is extremely unlikely that two slightly different messages will have the same hash.

A small change in a single word, “over” produces a major Change in the output; see https://en.wikipedia.org/wiki/Cryptographic_hash_function

Some Popular Cryptographic Hash Functions

- The **MD5** (message-digest) hash function is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number.
  - Invented by Ron Rivest of MIT in 1991; replaced the earlier MD4
- The **SHA-1, SHA-2** hash functions are also quite popular (160 bit, 20 byte value)
  - SHA-1 was broken in 2005; using SHA-2 family of algorithms is now favored, see
  - https://en.wikipedia.org/wiki/SHA-2
- **SHA-3**, released in 2015; it produces digests of size 224, 256, 384 and 512 bits
- **RIPEMD-160** – a family of cryptographic hash functions and so far has not been broken; produces a 160 bit (20 byte) digest
- **E.g. See Chrome, Settings, Security and Privacy, Security, Manage certificates, certificates, Verisign**

Copyright 2011-2022 Ellis Horowitz

Identifying Identical Web Pages – Four Approaches

1. ***Compare character by character*** two documents to see if they are identical
   - very time consuming !!
2. ***Hash just the first few characters and compare*** only those documents that hash to the same bucket
   - But what about web pages where every page begins with <HTML> ??
3. ***Use a hash function*** that examines the entire document
   - But this requires lots of buckets
4. ***Better approach*** - pick some fixed random positions for all documents and make the hash function depend only on these;
   - This avoids the problem of a common prefix for all or most documents, yet we need not examine entire documents unless they fall into a bucket with another document
   - But we still need a lot of buckets
5. ***Even better approach***: Compute the cryptographic hash (SHA-2 or MD5) of each web page and maintain in sorted order, *O(log n)* to search

Copyright 2011-2022 Ellis Horowitz

Identifying *Near* Identical Web Pages - Two Approaches

1. ***Produce fingerprints and test for similarity*** - Treat web documents as defined by a set of features, constituting an *n*-dimensional vector, and transform this vector into an *f-bit* fingerprint of a small size
   - Use Simhash or Hamming Distance to compute the fingerprint
     - SimHash is an algorithm for testing how similar two sets are
   - Compare fingerprints and look for a difference in at most k bits
   - E.g. see Manku et al., WWW 2007, *Detecting Near-Duplicates for Web Crawling,* http://www2007.org/papers/paper215.pdf
2. ***Instead of documents defined by n-vector of features, compute subsets of words (called shingles) and test for similarity of the sets***
   - Broder et al., WWW 1997, *Finding Near Duplicate Documents*

Copyright 2011-2022 Ellis Horowitz

General Paradigm

1. Define a function *f* that captures the contents of each document in a number
   - E.g. hash function, signature, or a fingerprint
2. Create the pair $<f(doc_i), ID\ of\ doc_i>$ for all doc_i
3. Sort the pairs
4. Documents that have the same *f*-value or an *f*-value within a small threshold are believed to be duplicates or near duplicates

Copyright 2011-2022 Ellis Horowitz

General Properties of Distance Measures

- **To compute similarity, we need a distance measure**
- **A distance measure must satisfy 4 properties**
  1. No negative distances
  2. *D(x,y) = 0 iff x=y*
  3. *D(x,y) = D(y,x)* symmetric
  4. *D(x,y) <= D(x,z) + D(z,y)* triangle inequality
- **There are several distance measures that can play a role in locating duplicate and near-duplicate documents**
  - *Euclidean distance* – D([x_1…x_n], [y_1,…,y_n]) = sqrt(Sum(x_i-y_i)^2) i=1…n
  - *Jaccard distance* – D(x,y) = 1 – SIM(x,y) or 1 minus the ratio of the sizes of the intersection and union of sets x and y
  - *Cosine distance* – the cosine distance between two points (two *n* element vectors) is the angle that the vectors to those points make; in the range 0 to 180 degrees
  - *Edit distance* – the distance between two strings is the smallest number of insertions and deletions of single characters that will convert one string into the other
  - *Hamming distance* – between two vectors is the number of components in which they differ (usually used on Boolean vectors)

Copyright 2011-2022 Ellis Horowitz

Set Distance and Set Similarity are Related

- **A set is an unordered collection of objects, e.g. {a, b, c}**
- **Focusing on the notion of *distance* of two sets we define a distance *d(A, B)* as**
  - *small,* if objects in A and B are close;
  - *large*, if objects in A and B are far apart;
  - *0,* if they are the same, and finally
  - *d(A, B)* is in the range [0, infinity]
- **Focusing on the notion of *similarity* we define *s(A, B)* as:**
  - *large,* if the objects in A and B are close;
  - *small,* if the objects in A and B are far apart;
  - *1,* if they are the same, and finally
  - *s(A, B)* is in the range [0, 1]
- Often we can convert between the two, as $d(A, B) = 1 - s(A, B)$

Copyright 2011-2022 Ellis Horowitz

Jaccard Similarity

- **Consider *A = {0, 1, 2, 5, 6}* and *B = {0, 2, 3, 5, 7, 9}***
- ***JS(A, B) = size(A intersection B) / size(A union B)***
- ***= size({0, 2, 5}) / size({0, 1, 2, 3, 5, 6, 7, 9})***
- ***= 3 / 8 = 0.375***
- **Suppose we divide our items into four clusters, e.g**
  - **$C_1 = \{0, 1, 2\}$**
  - **$C_2 = \{3, 4\}$**
  - **$C_3 = \{5, 6\}$**
  - **$C_4 = \{7, 8, 9\}$**

perhaps
C_1 represents action movies, C_2 comedies,
C_3 documentaries, C_4 horror movies

- **If $A_{clu} = \{C_1, C_3\}$ and $B_{clu} = \{C_1, C_2, C_3, C_4\}$, then**
- **$JS_{clu}(A, B) = JS(A_{clu}, B_{clu}) =$**
- **size (({C1, C3} intersect {C1, C2, C3, C4}) / ({C1, C3} union {C1, C2, C3, C4}))**
- **= 5 / 10 = 0.5**
- **If we are going to use Jaccard similarity to determine when two web pages are near duplicates; we need to say what are the elements of the sets we are comparing**

Copyright 2011-2022 Ellis Horowitz

Computing Jaccard Similarity from Sets Containing Shingles

- **Definition of Shingle**:
  - a contiguous subsequence of words in a document is called a *shingle;* The 4-shingling of the phrase below produces a bag of 5 items:

  "a rose is a rose is a rose" => a set S(D,w) is defined as

  { (a_rose_is_a), (rose_is_a_rose), (is_a_rose_is), (a_rose_is_a), (rose_is_a_rose)}
  - *S(D,w)* is the set of shingles of a document *D* of width *w*
- **Similarity Measures**
  - *Jaccard(A,B)* (also known as Resemblance) is defined as

    size of (S(A,w) intersect S(B,w)) / size of (S(A,w) union S(B,w))
  - *Containment(A,B)* is defined as

    size of (S(A,w) intersect S(B,w)) / size of (S(A,w))
  - 0 <= Resemblance <= 1
  - 0 <= Containment <= 1

- See *On the resemblance and containment of documents*, Conf. on Compression and Complexity, DEC Research Center, 1997

Copyright 2011-2022 Ellis Horowitz

Shingling Modeling Choices

- **White space?**
  - Should we include spaces and returns? Sometimes it makes sense, e.g.

    "plane has touch down" versus "threw a touchdown"

    (the space between "touch" and "down" is significant)
- **Capitalization?**
  - Sam versus sam. Can help to distinguish proper nouns
- **Punctuation?**
  - English is punctuated differently in the US and India; punctuation differs in articles, blogs, and tweets
- **How large should *k* be?**
  - General rule: high enough so the probability of almost all shingles matching is low, so a collision is meaningful;
- **Count replicas?**
  - Typically bag of words counts replicas, but shingling does not
- **Stop words?** Typically omitted as they are so common

Copyright 2011-2022 Ellis Horowitz

Mapping Shingles to Numbers

- **Original text**
  - "Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species"
- **All 3-shingles (there are 16 of them)**
  - (Tropical fish include), (fish include fish), (include fish found), (fish found in), (found in tropical), (in tropical environments), (tropical environments around), (environments around the), (around the world), (the world including), (world including both), (including both freshwater), (both freshwater and), (freshwater and salt), (and salt water), (salt water species)
- **Hash values for the 3-shingles (sets of shingles are large, so we hash them to make them more manageable, and we select a subset)**
  - 938, 664, 463, 822, 492, 798, 78, 969, 143, 236, 913, 908, 694, 553, 870, 779
- **Select only those hash values that are divisible by some number, e.g. here are selected hash values using *0 mod 4***
  - 664, 492, 236, 908; ***these are considered the fingerprints***
- **Near duplicates are found by comparing fingerprints and finding pairs with a high overlap**

Copyright 2011-2022 Ellis Horowitz

Jaccard Similarity and Shingling

- Recall the Jaccard similarity of sets A and B, *J(A,B)*, is defined as

  | *A intersect B* | / |*A union B* |
- The Jaccard distance of sets A and B, measuring *dis*similarity is defined as

  1 - J(A,B), or equivalently {(A union B) - (A intersect B)}/(A union B)
- We can test if two pages are near duplicate by
  1. First compute the *k*-shingles of the two pages
  2. Map the k-shingles into numbers (e.g. by hashing)
  3. Select a subset of the shingles to act as the fingerprints
  4. Compute the Jaccard similarity of the *k*-shingle fingerprints;
- **A high Jaccard similarity (e.g. greater than 0.9), implies the pages are near duplicate; or**
- ***if (J (fingerprint(A), fingerprint(B))) > k*, then the pages are similar;**

Copyright 2011-2022 Ellis Horowitz

Solution to Speed Up the Use of Shingles for Near Document Similarity

- If we have to compare the shingles of a newly found document with the fingerprints of all documents found so far, this process is still too time consuming
  - So we adopt a probabilistic approach to reduce the number of comparisons we must make
  - For each document *D* compute the sketch$_D$[*i*] is as follows:
    - Compute its shingles and then the hash values of those shingles
    - Map the hash values to $1..2^m$, call it *f(s)*
    - Let p_i be a *random permutation* on $1..2^m$
    - Pick the lowest bit, ie MIN $\{p_i(f(s))\}$ over all shingles *s* in *D*
    - *Do the above for 200 different permutations*
- Documents that share >= t (say 80%) corresponding vector elements are deemed to be near duplicates
- For more details see
- *Similarity estimation techniques from rounding algorithms* by Moses Charikar
- http://www.cs.princeton.edu/courses/archive/spring04/cos598B/bib/CharikarEstim.pdf

Copyright 2011-2022 Ellis Horowitz

Computing Sketch[i] for Doc1

Document 1

Start with 64-bit *f*(shingles)

Permute on the number line with π_i

Pick the min value

Test if Doc1.Sketch[i] = Doc2.Sketch[i]

Test for 200 random permutations: $\pi_1, \pi_2, \ldots \pi_{200}$

Doc1 (is near similar to) Doc2 iff the shingle with the MIN value in the union of Doc1 and Doc2 also lies in the intersection

Key Property: This happens with probability
Size_of_intersection / Size_of_union
which means this method is just as good as computing the complete Jaccard Similarity

Computing Near Duplicates Using Jaccard Similarity and Probability

- ***Theorem:*** $J(S(Doc_1), S(Doc_2)) = Prob(x^{\pi}_1 = x^{\pi}_2)$
- ***Said another way: the Jaccard similarity of the shingle sets of Doc1 and Doc2 is the same as the probability that the permutation of x1 equals the permutation of x2***

- Repeat the process independently for 200 random permutations π
- Call the set of the 200 resulting values of x^{π}_i the sketch $\psi(Doc_i)$ of Doc_i.
- We can then estimate the Jaccard coefficient for any pair of documents Doc_i and Doc_j to be
- $|\psi i \cap \psi j|/200;$
- if this exceeds a preset threshold, we declare that Doc_i and Doc_j are similar.

- See our textbook, section 19.6 for a proof

Copyright 2011-2022 Ellis Horowitz

Using SimHash to Detect Near Duplicates

- There is another way to determine if two web pages are near duplicates
- The method is called SimHash
- It was developed by Moses Charikar and is described in his paper *Similarity Estimation Techniques from Rounding Algorithms*, STOC May 2002
  - https://www.cs.princeton.edu/courses/archive/spring04/cos598B/bib/CharikarEstim.pdf
- The basic idea is the same as before
  - obtain an *f*-bit fingerprint for each document
  - A pair of documents are near duplicate if and only if fingerprints are at most *k*-bits apart
  - But in this case instead of using permutations and probability we use SimHash
- Documents D_1 and D_2 are near duplicates *iff*

  $Hamming\text{-}Distance(Simhash(D_1), Simhash(D_2)) <= K$
- Typically $f = 64$ and $k = 3$

Copyright 2011-2022 Ellis Horowitz

Simhash by Moses Charikar
A Locally Sensitive Hash Function

- **A hash function usually hashes different values to totally different hash values; here is an example**

  p1 = 'the cat sat on the mat'
  p2 = 'the cat sat on a mat'
  p3 = 'we all scream for ice cream'
  p1.hash => 415542861
  p2.hash => 668720516
  p3.hash => 767429688

- **Simhash is one where similar items are hashed to similar hash values (by similar we mean the bitwise Hamming distance between hash values is small)**

  p1.simhash => 851459198
  00110010110000000011110001111110
  p2.simhash => 847263864
  00110010100000000011100001111000
  p3.simhash => 984968088
  00111010101101010110101110011000

- **in this case we can see the hamming distance of the similar items (p1,p2)=4 is small whereas (p1,p3)=16 and (p2,p3)=12 are considerably larger**

Copyright 2011-2022 Ellis Horowitz

The Simhash Algorithm

- **The simhash of a phrase is calculated as follows:**

1. **pick a hashsize, lets say 32 bits**
2. **let V = [0] * 32 # (ie a vector of 32 zeros)**
3. **break the input phrase up into shingles, e.g.**
   'the cat sat on a mat'.shingles(2) =>
   #<Set: {"th", "he", "e ", " c", "ca", "at", "t ", " s", "sa", “at”, “t “, " o", "on", "n ", " a", “a “, " m", "ma”, “at”}>
4. **hash each feature using a normal 32-bit hash algorithm (MD5 or SHA)**
   "th".hash = -502157718 "he".hash = -369049682 ...
5. **for each hash**
   if bit_i of hash is set then add 1 to V[i]
   if bit_i of hash is not set then subtract 1 from V[i]
6. **simhash bit_i is 1 if V[i] > 0 and 0 otherwise**

- **Simhash is useful because if the Simhash bitwise Hamming distance of two phrases is low then their Jaccard coefficient is high**

Copyright 2011-2022 Ellis Horowitz

Simhash Continued

- **In the case that two numbers have a low bitwise Hamming distance and the difference in their bits are in the lower order bits then it turns out that they will end up close to each other if the list is sorted.**
- **consider the eight numbers and their bit representations**
- **1 37586 1001001011010010**
- **2 50086 1100001110100110 7 <--(this column lists hamming**
- **3 2648 0000101001011000 11 distance to previous entry)**
- **4 934 0000001110100110 9**
- **5 40957 1001111111111101 9**
- **6 2650 0000101001011010 9**
- **7 64475 1111101111011011 7**
- **8 40955 1001111111111011 4**

if we sort them

4	934	0000001110100110	
3	2648	0000101001011000	9
6	2650	0000101001011010	1
1	37586	1001001011010010	5
8	40955	1001111111111011	6
5	40957	1001111111111101	2
2	50086	1100001110100110	9
7	64475	1111101111011011	9

notice that two pairs with very smallest hamming distance hdist(3,6)=1 and hdist(8,5)=2 have ended up adjacent to each other.

Copyright 2011-2022 Ellis Horowitz

Simhash Continued

- **Rather than check every combo we could just check the adjacent pairs of the list, each is a good candidate.**
- **This reduces the runtime from $n*(n-1)/2$ coefficient calculations, $O(n^2)$ to**
  - **n fingerprints calculations $O(n)$ +**
  - **a sort $O(n \log n)$ +**
  - **n coefficient calculations $O(n)$,**
- **which is $O(n \log n)$ overall;**

- **A problem:**
  - **there is another pair with a low Hamming distance, $hdist(4,2)=2$ that have ended up totally apart at other ends of the list...**
  - **sorting only picked up the pairs that differed in their lower order bits.**

Copyright 2011-2022 Ellis Horowitz

Simhash Continued

- To get around this consider another convenient property of bitwise Hamming distance, *a permutation of the bits of two numbers preserves Hamming distance*
- If we permute by 'rotating' the bits, i.e. bit shift left and replace lowest order bit with the 'lost' highest order bit we get 'new' fingerprints that have the same Hamming distances

'rotate' bits left twice
4 3736 0000111010011000
3 10592 0010100101100000 9
6 10600 0010100101101000 1
1 19274 0100101101001010 5
8 32750 0111111111101110 6
5 32758 0111111111110110 2
2 3739 0000111010011011 9
7 61295 1110111101101111 9

if we sort again by fingerprint
4 3736 0000111010011000
2 3739 0000111010011011 2
3 10592 0010100101100000 11
6 10600 0010100101101000 1
1 19274 0100101101001010 5
8 32750 0111111111101110 6
5 32758 0111111111110110 2
7 61295 1110111101101111 6

this time the (2,4) pair ended up adjacent
we also identified the (3,6) and (5,8) pairs as candidates again

Copyright 2011-2022 Ellis Horowitz

Simhash Continued

- So to avoid whether the items differ in the higher or lower order bits we can just do the following B times (where B is the bit length of the fingerprint)

1. rotate the bits
2. sort
3. check adjacent

- in the likely case that B << n the entire algorithm will still be under $O(n^2)$
- (in fact we wouldn't check each adjacent pair each rotation, rather collect the adjacent pairs in a set and check at the end, very similar items would end up adjacent in a number of the rotations but only need to be checked once)
- but it all hinges on the fact that similar items end up next to each other in the sorted lists

- http://matpalm.com/resemblance/simhash/
- http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/33026.pdf

Copyright 2011-2022 Ellis Horowitz

HW2 Assignment

- Involves
  1. Java programming
     - I assume all of you know how to program in Java!
  2. Eclipse Software Development Environment
  3. crawler4j, an open source java web crawler
  4. a crawl and analysis of a web site and an analysis of the crawl

What is Eclipse?

- Eclipse started as a proprietary IBM product (IBM Visual age for Smalltalk/Java)
  - Embracing the open source model IBM opened the product up
- Open Source
  - It is a general purpose open platform that facilitates and encourages the development of third party plug-ins
- Best known as an Integrated Development Environment (IDE)
  - Provides tools for coding, building, running and debugging applications
- Originally designed for Java, now supports many other languages
  - Good support for C, C++
  - Python, PHP, Ruby, etc…

Prerequisites for Running Eclipse

- Eclipse is written in Java and will thus need an installed JRE (Java Runtime Environment) or JDK (Java Development Kit) in which to execute
  - JDK recommended

Obtaining Eclipse

- Eclipse can be downloaded from…
  https://www.eclipse.org/downloads/packages/
- Eclipse comes bundled as a zip file (Windows) or a tarball (all other operating systems)
- Eclipse version to Install - Eclipse IDE for Java Developers

Installing Eclipse

- Simply unwrap the zip file to some directory where you want to store the executables
- The document

"Instructions for Installing Eclipse and crawler4j"

- located at

http://csci572.com/2022Fall/hw2/Crawler4jinstallation.pdf

describes the installation for both Windows and Macs

Launching Eclipse

- Once you have the environment setup, go ahead and launch eclipse
- You should see a splash screen such as the one below

Eclipse Structure

» The following components constitute the rich client platform:

- Core platform - boot Eclipse, run plug-ins
- OSGi - a standard bundling framework
- the Standard Widget Toolkit (SWT) - a portable widget toolkit
- JFace - file buffers, text handling, text editors
- The Eclipse Workbench - views, editors, perspectives, wizards

Eclipse Platform is the common base
Consists of several key components

Eclipse Overview

Selecting a Workspace

- In Eclipse, all of your code will live under a *workspace*
- A *workspace* is nothing more than a location where we will store the source code and where Eclipse will write out preferences
- Eclipse allows you to have multiple workspaces – each tailored in its own way
- Choose a location where you want to store your files, then click OK

Copyright Ellis Horowitz

Workspace Component

- Tools operate on files in user's **workspace**
- Workspace holds 1 or more top-level **projects**
- Projects map to directories in file system
- Tree of **folders** and **files**
- {Files, Folders, Projects} termed **resources**

- Tools read, create, modify, and delete resources in workspace
- Plug-ins access via workspace and resource APIs

Eclipse IDE Components

Menubars

Full drop down menus plus quick access to common functions

Perspective Switcher

We can switch between various perspectives here

Task List Pane

This contains a list of "tasks" to complete

Editor Pane

This is where we edit our source code

Package Explorer Pane

This is where our projects/files are listed

Outline Pane

This contains a hierarchical view of a source file

Miscellaneous Pane

Various components can appear in this pane – typically this contains a console and a list of compiler problems

Copyright Ellis Horowitz

Creating a New Project

- All code in Eclipse needs to live under a project
- To create a project: File → New → Java Project

Copyright Ellis Horowitz

Creating a New Project (continued)

- Enter a name for the project, then click Finish

Hello-world Project

Copyright Ellis Horowitz

Creating a New Project (continued)

- The newly created project should then appear under the Package Explorer

Copyright Ellis Horowitz

The src folder

- Eclipse automatically creates a folder to store your source code in called src

Copyright Ellis Horowitz

Creating a Class

- To create a class, simply click on the New button, then select Class

Copyright Ellis Horowitz

Creating a Class (continued)

- This brings up the new class wizard
- From here you can specify the following...
  - Package
  - Class name
  - Superclass
  - Whether or not to include a main
  - Etc…
- Fill in necessary information then click Finish to continue

The Created Class

- As you can see a number of things have now happened…

Directory structure for package and actual java file created automatically

Source is loaded into the editor pane, already stubbed out

Source displayed in a hierarchical fashion listing each method name

Copyright Ellis Horowitz

Compiling Source Code

- One important feature of Eclipse is that it automatically compiles your code in the background
- This means that errors can be corrected when made
  - We all know that iterative development is an excellent approach to developing code, but going to shell to do a compile can interrupt the normal course of development
  - You no longer need to go to the command prompt and compile code directly

Example Compilation Error

- This code contains a typo in the println statement…

Packages/Classes with errors are marked with a red X

Often Eclipse may have suggestions on how to fix the problem – if so, a small light bulb will be displayed next to the line of offending code

Error underlined with red squiggly line (just like spelling errors in many word processors)

Position in file is marked with a red line – 1 click allows you to jump to line with error

Methods with errors are marked with a red X

The Problems tab will contain a full representation of all errors across all files of all open projects

Copyright Ellis Horowitz

Running Code

- An easy way to run code is to right click on the class and select Run As → Java Application

Copyright Ellis Horowitz

Running Code (continued)

- The output of running the code can be seen in the Console tab in the bottom pane

Copyright Ellis Horowitz

Eclipse Java Debugger

Run or debug Java programs

Local variables

Threads and stack frames

Editor with breakpoint marks

Console I/O

Debugging Code

- Eclipse comes with a pretty good built-in debugger
- You can set breakpoints in your code by double clicking in the left hand margin – break points are represented by these blue bubbles

Copyright Ellis Horowitz

End of Eclipse Tutorial

Tools for Surface Web Crawling

- Command line for issuing http requests
  - wget, pre-installed in Ubuntu
    - get a single page
    - wget http://www.example.com/index.html
    - support http, ftp etc., e.g.
    - wget ftp://ftp.gnu.org/pub/gnu/wget/wget-latest.tar.gz
  - curl, OSX pre-installed also supports http requests
- Simple crawling programs
  - Crawler4j, written in Java
  - Scrapy: http://scrapy.org, written in Python
- Large-scale crawling programs
  - Heritrix, crawler for archive.org
  - Nutch, Apache Software Foundation

How To Get a Web Page in Java

```
import java . net .*;
import java . io .*;
public class URLReader {
public static void main(String [] args) throws Exception { } }
  URL oracle = new URL(”http://www.oracle.com/”);
  BufferedReader in = new BufferedReader (
  new InputStreamReader(oracle.openStream()));
  String inputLine ;
  while (( inputLine = in . readLine ()) != null)
     System . out . println ( inputLine );
     in . close ();
   }
}
```

- After you create a URL, you can call the URL's openStream() method to get a stream from which you can read the contents of the URL.
- The openStream() method returns a java.io.InputStream object

Instructions for Installing Crawler4j

- download crawler4j from github
  - GitHub is a web-based repository hosting service for software. Originally the Git system offered distributed revision control and source code management (SCM) functionality, but on the command line; GitHub offers a web interface and some additional features.
  - As of January 2020, GitHub reports having over 56 million users and over 100 million repositories
  - Microsoft purchased GitHub

Downloading Crawler4j from GitHub

https://github.com/yasserg/crawler4j/

See especially the README file page at
https://github.com/yasserg/crawler4j/blob/master/README.md

Copyright Ellis Horowitz

Crawler4j Source Code

Copyright Ellis Horowitz

Crawler folder, a good place to start; look especially at WebCrawler.java

Crawler4j Source code is Logically Organized into folders

Fetcher Code handles:
- schemes: http, https
- politeness delay;
- redirects;
- max-size settings;
- expired connections

Frontier Code handles:
- statistics database;
- previously seen URLs
- queue of pending URLs

Copyright Ellis Horowitz

Crawler4j Routines are Named According to their Function

Parser Code handles:
- binary data
- html pages
- extracting links

Robots.txt Code handles:
- fetching and re-fetching robots.txt
- caching robots.txt files
- interpreting commands
- working with Page Fetcher

Copyright Ellis Horowitz

More crawler4j Source code

utility routines

URL resolver and canonicalizer handles:
- checking against list of TLDs
- normalizes URL, removes . or .., etc
- alters name/value pairs
- converts #nn values
- evaluates <base>

Copyright Ellis Horowitz

Logic Flowchart

http://csci572.com/2022Fall/hw2/Crawler4JProcess.pdf

Copyright Ellis Horowitz

Configuring the Crawler and Seeding it

```
public class Controller {
   public static void main(String[] args) throws Exception {
      String crawlStorageFolder = "/data/crawl";
      int numberOfCrawlers = 7;
      CrawlConfig config = new CrawlConfig();
      config.setCrawlStorageFolder(crawlStorageFolder);
      /*  Instantiate the controller for this crawl.*/
      PageFetcher pageFetcher = new PageFetcher(config);
      RobotstxtConfig robotstxtConfig = new RobotstxtConfig();
      RobotstxtServer robotstxtServer = new RobotstxtServer(robotstxtConfig, pageFetcher);
      CrawlController controller = new CrawlController(config, pageFetcher, robotstxtServer);
      /*  For each crawl, you need to add some seed urls. These are the first
       * URLs that are fetched and then the crawler starts following links
       * which are found in these pages  */
      controller.addSeed("https://www.nytimes.com/");
      /*  Start the crawl. This is a blocking operation, meaning that your code
       * will reach the line after this only when crawling is finished.  */
      controller.start(MyCrawler.class, numberOfCrawlers);
   }
}
```

folder to store downloads;

#crawlers

set up pagefetcher and robots.txt handler

crawling www.nytimes.com

Crawling Pages Other Than HTML

- To make sure you are not just crawling HTML, and missing pdf and doc files, you need to set BinaryContent to true
- Examine the routine BasicCrawlController and see line 31
- Turn config.setIncludeBinaryContentInCrawling(false);

https://github.com/yasserg/crawler4j/blob/master/crawler4j-examples/crawler4j-examples-base/src/test/java/edu/uci/ics/crawler4j/examples/basic/BasicCrawlController.java

Defining Which Pages to Crawl

```
public class MyCrawler extends WebCrawler {
   private final static Pattern FILTERS =
Pattern.compile(".*(\\.(css|js|gif|jpg"  + "|png|mp3|mp3|zip|gz))$");
  /** This method receives two parameters. The first parameter is the page
    * in which we have discovered this new url and the second parameter is
    * the new url. You should implement this function to specify whether
    * the given url should be crawled or not (based on your crawling logic).
    * In this example, we are instructing the crawler to ignore urls that
    * have css, js, git, ... extensions and to only accept urls that start
    * with "http://www.latimes.com/". In this case, we didn't need the
    * referring Page parameter to make the decision.   */
    @Override
    public boolean shouldVisit(Page referringPage, WebURL url) {
        String href = url.getURL().toLowerCase();
        return !FILTERS.matcher(href).matches()
                && href.startsWith("http://www.nytimes.com/");
    }
```

see next slide

Matching URLs

- ".*(\\.(css|js|gif|jpg" + "|png|mp3|mp4|zip|gz))$"

- A regular expression, specified as a string, must first be compiled into an instance of this class.
- a Matcher object that can match arbitrary character sequences against the regular expression
- See https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html

- In the above there are two strings concatenated by plus; consider the simpler form:
- ".*(\\.(css|js|zip|gz))$"

- . matches any character
- * matches zero or more of preceding character
- \. matches a literal dot
- $ anchors the pattern at the end of the string

Parsing the Downloaded Page

```
/**   This function is called when a page is fetched and ready
     * to be processed by your program.  */
    @Override
    public void visit(Page page) {
        String url = page.getWebURL().getURL();
        System.out.println("URL: " + url);
        if (page.getParseData() instanceof HtmlParseData) {
            HtmlParseData htmlParseData = (HtmlParseData) page.getParseData();
            String text = htmlParseData.getText();
            String html = htmlParseData.getHtml();
            Set<WebURL> links = htmlParseData.getOutgoingUrls();
            System.out.println("Text length: " + text.length());
            System.out.println("Html length: " + html.length());
            System.out.println("Number of outgoing links: " + links.size());
        }
   }
```

The Actual Exercise

- the URLs it attempts to fetch, fetch.csv. The number of rows should be no more than 20,000 as that is our pre-set limit.
- the files it successfully downloads, visit.csv; clearly the number of rows will be less than the number of rows in fetch.csv
- all of the URLs that were discovered and processed in some way; urls.csv. This file could be much larger than 20,000 rows as it will have numerous repeated URLs

Copyright Ellis Horowitz

Things to Save

- Fetch statistics:
  - # fetches attempted:
    The total number of URLs that the crawler attempted to fetch. This is usually equal to the MAXPAGES setting if the crawler reached that limit; less if the website is smaller than that.
  - # fetches succeeded:
    The number of URLs that were successfully downloaded in their entirety, i.e. returning a HTTP status code of 2XX.
  - # fetches failed or aborted:
    The number of fetches that failed for whatever reason, including, but not limited to: HTTP redirections (3XX), client errors (4XX), server errors (5XX) and other network-related errors.
-

Outgoing URLs

- Outgoing URLs: statistics about URLs extracted from visited HTML pages
  - Total URLs extracted:
    The grand total number of URLs extracted from all visited pages
  - # unique URLs extracted:
    The number of unique URLs encountered by the crawler
  - # unique URLs within the news web site:
    The number of unique URLs encountered that are associated with the news website,
    i.e. the URL begins with the given root URL of the news website.
  - # unique URLs outside the news website:
    The number of unique URLs encountered that were not from the website.

Copyright Ellis Horowitz

Sample Crawl Report for NY Times Using 20,000 as the Download Limit

```
News site crawled: https://www.nytimes.com/
Fetch Statistics
===================
# fetches attempted:19750
# fetches succeeded:19067
# fetches aborted or failed:683
Outgoing URLs
===================
Total URLs extracted:964461
# unique URLs extracted:73450
# unique URLs within nytimes:31115
# unique URLs outside nytimes:42335
Status Codes
===================
200 OK:19067
301 Moved Permanently:140
302 Moved Temporarily:21
503 Service Unavailable:366
401 Unauthorized Error:1
403 Forbidden:8
404 Not Found:143
500 Internal Server Error:1
400 Bad Request:3
File Sizes
===================
<1KB:14
1KB~<10KB:1052
10KB~<100KB:694
100KB~<1MB:8602
>=1MB:1
Content Type
===================
text/html:18349
image/jpeg=5
image/png=3
```

Sample Fetch File for NY Times

	A	B
2	https://www.nytimes.com/	200
3	https://www.nytimes.com/video	200
4	https://www.nytimes.com/section/arts/dance	200
5	https://www.nytimes.com/section/us	200
6	https://www.nytimes.com/svc/collections/v1/publish/https:/www.nytimes.com/se	200
7	https://www.nytimes.com/section/technology	200
8	https://www.nytimes.com/svc/collections/v1/publish/https:/www.nytimes.com/se	200
9	https://www.nytimes.com/column/speakingindance	200
10	https://www.nytimes.com/live/2020/impeachment-trial-live-01-27/the-house-want	200
11	https://www.nytimes.com/svc/collections/v1/publish/https:/www.nytimes.com/se	200
12	https://www.nytimes.com/video/investigations	200
13	https://www.nytimes.com/subscription?campaignId=37WXW	200
14	https://www.nytimes.com/section/science	200
15	https://www.nytimes.com/section/business/smallbusiness	200
16	https://www.nytimes.com/svc/collections/v1/publish/https:/www.nytimes.com/co	200
17	https://www.nytimes.com/svc/collections/v1/publish/https:/www.nytimes.com/se	200
18	https://www.nytimes.com/section/food	200
19	https://www.nytimes.com/news-event/2020-election	200
20	https://www.nytimes.com/live/2020/impeachment-trial-live-01-27/starr-manages-t	200
21	https://www.nytimes.com/subscription	200
22	https://www.nytimes.com/live/2020/impeachment-trial-live-01-27/rudy-giuliani-is-	200

Copyright Ellis Horowitz

Sample Visit File for NY Times

URL	Size	Outgoing Links	Content Type
https://www.nytimes.com/	1451085	139	text/html
https://www.nytimes.com/video	429553	109	text/html
https://www.nytimes.com/section/arts/dance	736631	157	text/html
https://www.nytimes.com/section/us	946441	120	text/html
https://www.nytimes.com/section/technology	1112691	161	text/html
https://www.nytimes.com/column/speakingindance	459835	141	text/html
https://www.nytimes.com/live/2020/impeachment-trial-live-01-27/the-h	314892	149	text/html
https://www.nytimes.com/subscription?campaignId=37WXW	72384	18	text/html
https://www.nytimes.com/video/investigations	1247877	166	text/html
https://www.nytimes.com/section/science	1012715	182	text/html
https://www.nytimes.com/section/business/smallbusiness	817605	141	text/html
https://www.nytimes.com/section/food	1092598	173	text/html
https://www.nytimes.com/news-event/2020-election	1338179	173	text/html
https://www.nytimes.com/subscription	72384	18	text/html
https://www.nytimes.com/live/2020/impeachment-trial-live-01-27/starr	256830	136	text/html
https://www.nytimes.com/live/2020/impeachment-trial-live-01-27/rudy	309304	148	text/html
https://www.nytimes.com/newsletters/signup/CN	107087	92	text/html
https://www.nytimes.com/subscription/education?campaignId=7KL9U	93183	19	text/html
https://www.nytimes.com/live/2020/impeachment-trial-live-01-27/trum	339037	153	text/html
https://www.nytimes.com/section/food/drinks	811026	166	text/html

Copyright Ellis Horowitz

What to Submit

- Compress all of the above into a single zip archive and name it:

  crawl.zip
- Use only standard zip format. Do NOT use other formats such as zipx, rar, ace, etc. For example the zip file might contain the following three files:

1. CrawlReport_nytimes.txt,
2. fetch_nytimes.csv
3. visit_nytimes.csv

- Place your crawl.zip file in your csci572/hw2 folder

Introduction to Information Retrieval

Copyright Ellis Horowitz 2011-2022

Information Retrieval

- ***Information retrieval (IR)* has been a computer science subject for many decades**
  - Traditionally it deals with the *indexing* of a given set of textual documents and the *retrieval* of relevant documents given a query
- Searching for pages on the World Wide Web has become the "killer app."

- **There has been a great deal of research on**
  - How to index a set of documents (a corpus)
  - How to efficiently retrieve relevant documents
- Jurafsky and Manning have an excellent video introducing the subject of Information Retrieval;
- http://csci572.com/movies/01_IntroIR.mp4 (9 minutes)

then jump to slide 16

Copyright Ellis Horowitz, 2011-2022

The Traditional IR System

Copyright Ellis Horowitz 2011-2022

History of IR

- **1960-70's:**
  - Initial exploration of text retrieval systems for "small" corpora of scientific abstracts, and law and business documents.
  - Development of the basic Boolean and vector-space models of retrieval.
  - Prof. Salton and his students at Cornell University were the leading researchers in the area.

Copyright Ellis Horowitz, 2011-2022

IR History Continued

- **1980's:**
  - **Creation of large document database systems, many run by companies:**
    - Lexis-Nexis, http://www.lexisnexis.com/
      - information to legal, corporate, government and academic markets, and publishes legal, tax and regulatory information
    - Dialog, http://www.dialog.com/
      - data from more than 1.4 billion unique records of key information.
    - MEDLINE, http://www.medlineplus.gov/
      - National Library of Medicine health information

Copyright Ellis Horowitz, 2011-2022

IR History Continued

- **1990's:**
  - **Searching FTP'able documents on the Internet**
    - Archie
    - WAIS
  - **After the World Wide Web is invented, search engines appear**
    - Lycos
    - Yahoo
    - Altavista

Copyright Ellis Horowitz, 2011-2022

IR History Continued

- **1990's continued:**
  - **Organized Competitions**
    - NIST TREC (Text REtrieval Conferences, http://trec.nist.gov/)
    - Sponsored by National Institute of Standards and Technology, NIST
  - **Several New Types of IR Systems are Developed**

    1. *Recommender Systems*: computer programs which attempt to predict items (movies, music, books, news, web pages) that a user may be interested in, given some information about the user's profile.
       - **Often implemented as a collaborative filtering algorithm, examples include:**
         - » **YouTube, perhaps the largest scale such system in existence**
         - » **https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45530.pdf**
         - » **Amazon's recommendation system, see https://stackoverflow.com/questions/2323768/how-does-the-amazon-recommendation-feature-work**
         - » **http://rejoiner.com/resources/amazon-recommendations-secret-selling-online/**

    2. *Automated Text Categorization & Clustering Systems*
       - **Useful for grouping news articles**

Copyright Ellis Horowitz, 2011-2022

Recent IR History Moves to the Web

- **2000's**
  - **Link analysis for Web Search**
    - Google started this
  - **Extension to retrieval of multimedia: images, music, video**
    - It is much harder to index multimedia artifacts
  - **Question Answering**
    - Question answering systems return an actual answer rather than a ranked list of documents
    - Since 1999 TREC has had a Question/Answer track, see http://trec.nist.gov/data/qa.html

Copyright Ellis Horowitz, 2011-2022

Areas Related To, But Different Than, Information Retrieval

- Database Management
- Library and Information Science
- Artificial Intelligence
- Natural Language Processing
- Machine Learning
- Data Science

Copyright Ellis Horowitz, 2011-2022

Database Management is Different from IR

- Focused on *structured* data stored in relational tables rather than free-form text
- Focused on efficient processing of well-defined queries in a formal language (SQL)
- Clearer semantics for both data and queries
- Web pages are mostly unstructured, though the Document Object Model (DOM) can provide some clues

Copyright Ellis Horowitz, 2011-2022

- **Focused on the human user aspects of information retrieval (human-computer interaction, user interface, visualization).**
- **Concerned with effective categorization of human knowledge.**
- **Concerned with citation analysis and *bibliometrics* (structure of information).**
- **Recent work on *digital libraries* brings it closer to Computer Science & IR.**

Copyright Ellis Horowitz, 2011-2022

Artificial Intelligence

- **Focused on the representation of knowledge, reasoning, and intelligent action.**
- **Formalisms for representing knowledge and queries:**
  - ***First-order Predicate Logic*** – a formal system that uses quantified variables over a specified domain of discourse
  - ***Bayesian Networks*** – a directed acyclic graph model that represents a set of random variables and their dependencies
    - **E.g.** A Bayesian Network that represents the probabilistic relationships between diseases and symptoms
- **Recent work on web ontologies and intelligent information agents brings it closer to IR**
  - Web Ontology Language OWL is a family of knowledge representation languages for authoring ontologies
  - See https://www.w3.org/OWL/

Copyright Ellis Horowitz, 2011-2022

Natural Language Processing

- Focused on the syntactic, semantic, and pragmatic analysis of natural language text and discourse.
- Ability to analyze syntax (phrase structure) and semantics allows retrieval based on *meaning* rather than keywords
- NLP now uses vast amounts of web pages as data to run statistical and machine learning models to infer meaning

Copyright Ellis Horowitz, 2011-2022

Machine Learning

- **A branch of Artificial Intelligence concerned with algorithms that allow computers to evolve their behavior based on empirical data**
- **Focused on the development of computational systems that improve their performance with experience**
- **Two major subtypes of machine learning are**:
  - Automated classification of examples based on learning concepts from labeled training examples (*supervised learning*).
  - Automated methods for clustering unlabeled examples into meaningful groups (*unsupervised learning*)
- **Machine learning is distinct from *data mining*, which focuses on the discovery of previously unknown properties of the given data**
  - Data mining is akin to query analysis and ranking

Copyright Ellis Horowitz, 2011-2022

Data Science

- "*Data science* is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structural and unstructured data. Data science is related to data mining, machine learning and big data." wikipedia
- A *data scientist* is someone who knows how to extract meaning from and interpret data, which requires both tools and methods from **statistics** and **machine learning**, as well as human review. They spend a lot of time in the process of collecting and cleaning data, because data is never clean
- For fun watch *The Beauty of Data Visualization*
- **https://www.youtube.com/watch?v=5Zg-C8AAIGg (18 min)**

Copyright Ellis Horowitz, 2011-2022

Basic Information Retrieval Begins with Keyword Matching

- **Simplest notion of relevance is that the query string appears verbatim in the document.**
  - Slightly less strict notion is that the words in the query appear frequently in the document, in any order (this is like viewing the document as a *bag of words*).
- **But that may not retrieve relevant documents that include synonymous terms.**
  - "restaurant" vs. "café"
  - "PRC" vs. "China"
- **And it may retrieve irrelevant documents that include ambiguous terms.**
  - "bat" (baseball vs. mammal)
  - "Apple" (company vs. fruit)
  - "bit" (unit of data vs. act of eating)

Copyright Ellis Horowitz, 2011-2022

- **Goes beyond using just keyword matching, instead it**
  - Takes into account the *meaning* of the words used
  - Takes into account the *order* of words in the query
  - Adapts to the user based on direct or indirect feedback
  - Takes into account the *authority* of the source

Copyright Ellis Horowitz, 2011-2022

A More Detailed IR Architecture

Logical View

- User needs are part of the input
- User feedback is provided
- Queries initiate a search of the index and docs are retrieved
- A ranking function orders the results

Start with a text database; it is indexed; a user interface permits query operations which cause a search on the Index; matched documents are retrieved and ranked

Copyright Ellis Horowitz 2011-2022

Defining Terms

- **Parsing forms index words (tokens) and includes:**
  - *Stopword* removal
    - See http://www.ranks.nl/stopwords for google stopwords
  - *Stemming*: reducing a word to its root
    - More about this later
- ***Indexing* constructs an *inverted index* of word to document pointers.**
- ***Searching* retrieves documents that contain a given query token from the inverted index.**
- ***Ranking* scores all retrieved documents according to a relevance metric.**

Copyright Ellis Horowitz, 2011-2022

Boolean and Vector Space Retrieval Models

***Primary goal*: to formalize the processes that underlie a person making the decision that a piece of text is relevant to his information need**

Copyright Ellis Horowitz 2011-2022

Retrieval Models

- **A retrieval model specifies the details of:**
  - Document representation
  - Query representation
  - Retrieval function
- **Determines a notion of relevance.**
- **Notion of relevance can be binary or continuous (i.e. *ranked retrieval*)**
- **Three major Information Retrieval Models are:**

1. Boolean models (set theoretic) (*Chapter 1 in Manning et al*)
2. Vector space models (statistical/algebraic) (*Chapter 2 in Manning et al*)
3. Probabilistic models (*Chapter 11 in Manning et al*)

Copyright Ellis Horowitz, 2011-2022

Common Pre-Processing Steps

1. Strip unwanted characters/markup (e.g. HTML tags, punctuation, page numbers, etc.).
2. Break into tokens (keywords) separating out whitespace.
3. Stem tokens to "root" words

4. Remove common stopwords (e.g. a, the, it, etc.).
5. Detect common phrases (possibly using a domain specific dictionary).
6. Build inverted index (keyword → list of docs containing it).

Copyright Ellis Horowitz, 2011-2022

Boolean Model

- **A document is represented as a set of keywords.**
- **Queries are Boolean expressions of keywords, connected by AND, OR, and NOT, including the use of brackets to indicate scope**
- **Here is a sample Boolean query with explicit AND, OR, NOT operators**
  - [[Rio & Brazil] | [Hilo & Hawaii]] & hotel & !Hilton]

Google Advanced Search;
Note inclusion of AND, OR, NOT operators

Copyright Ellis Horowitz, 2011-2022

Boolean Retrieval Model

- **Popular retrieval model because:**
  - Easy to understand for simple queries.
  - Clean formalism.
- **Boolean models can be extended to include ranking**
- **Reasonably efficient implementations possible for normal queries.**

Copyright Ellis Horowitz, 2011-2022

Boolean Models – Problems

- **Very rigid: AND means all; OR means any**
- **Difficult to express complex user requests**
- **Difficult to control the number of documents retrieved**
  - *All* matched documents will be returned
- **Difficult to rank output**
  - *All* matched documents logically satisfy the query
- **Difficult to perform relevance feedback**
  - If a document is identified by the user as relevant or irrelevant, how should the query be modified?

Copyright Ellis Horowitz, 2011-2022

Problem Example For the Boolean Model

- **The simple query "Lincoln"**
  - Too many matches including Lincoln cars and places named Lincoln as well as Abraham Lincoln
- **More detailed query "President AND Lincoln"**
  - Returns documents that discuss the President of Ford Motor company that makes the Lincoln car
- **Even more detailed query "president AND Lincoln AND NOT (automobile OR car)"**
  - Better, but the use of NOT will remove a document about President Lincoln that says "Lincoln's body departs Washington in a nine car funeral train"
- **Perhaps try**
  - President AND lincoln AND biography AND life AND birthplace AND gettysburg AND NOT (automobile OR car), but too many ANDs can lead to nothing, so
  - President AND lincoln AND (biography OR life OR birthplace OR gettysburg) AND NOT (automobile OR car)

Copyright Ellis Horowitz, 2011-2022

The Vector-Space Model

- **Assume *t* distinct terms remain after preprocessing; call them index terms or the vocabulary**
- **These "orthogonal" terms form a vector space**

  size of the vocabulary = Dimension = t = |vocabulary|
- **A document D_i is represented by a vector of index terms**

  $$D_i = (d_{i1}, d_{i2}, \ldots, d_{it})$$
- **Where d_{ij} represents the weight of the *j*-th term in the ith doc**
  - *but how is the weight computed?*
- **Both documents and queries are expressed as *t*-dimensional vectors**

Copyright Ellis Horowitz, 2011-2022

Graphic Representation Example

Example:

$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = 0T_1 + 0T_2 + 2T_3$

Vocabulary consists of 3 terms with weights the coefficients
There are two documents, D_1 and D_2; there is one query, Q

- Is D_1 or D_2 more similar to Q?
- How to measure the degree of similarity? Distance? Angle? Projection?

Copyright Ellis Horowitz, 2011-2022

Representing a Collection of Documents

- A collection of *n* documents can be represented in the vector space model by a term-document matrix.
- An entry in the matrix corresponds to the "weight" of a term in the document; zero means the term has no significance in the document or it simply doesn't exist in the document; but we still need a way to compute the weight

$$
\begin{array}{c|cccc}
 & T_1 & T_2 & \cdots & T_t \\
\hline
D_1 & w_{11} & w_{21} & \cdots & w_{t1} \\
D_2 & w_{12} & w_{22} & \cdots & w_{t2} \\
\vdots & \vdots & \vdots & & \vdots \\
\vdots & \vdots & \vdots & & \vdots \\
D_n & w_{1n} & w_{2n} & \cdots & w_{tn}
\end{array}
$$

Copyright Ellis Horowitz, 2011-2022

Term Weights: Term Frequency

- **One way to compute the weight is to use the term's frequency in the document**
- **Assumption: the more frequent terms in a document are more important, i.e. more indicative of the topic.**

$$f_{ij} = \text{frequency of term } i \text{ in document } j$$

- **May want to normalize *term frequency* (*tf*) across the entire corpus:**

$$tf_{ij} = f_{ij} \ / \ max\{f_{ij}\}$$

Copyright Ellis Horowitz, 2011-2022

Term Weights: Inverse Document Frequency

- Terms that appear in many *different* documents are *less* indicative of overall topic

  df_i = document frequency of term i

  = number of documents containing term i

of course df_i is always <= N (total number of documents)

idf_i = inverse document frequency of term i,

$= \log_2 (N / df_i)$

(N: total number of documents)

- An indication of a term's *discrimination* power
- Log is used to dampen the effect relative to *tf*

Copyright Ellis Horowitz, 2011-2022

An Example of Inverse Document Frequency

term	df_i	idf_i
Calpurnia	1	log(1,000,000/1)=6
animal	100	4
Sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

- $idf_i = \log_{10}(N/df_i)$, $N = 1{,}000{,}000$
- there is one *idf* value for each term *t* in a collection

Copyright Ellis Horowitz, 2011-2022

TF.IDF Weighting

- **A typical combined term importance indicator is *tf-idf weighting* (*note: it is often written with a hyphen, but the hyphen is NOT a minus sign; some people replace the hyphen with a dot*):**

$$w_{ij} = tf_{ij}.idf_i = (1 + \log tf_{ij}) * \log_2 (N/ df_i)$$

- **A term occurring frequently in the document but rarely in the rest of the collection is given high weight.**
- **Many other ways of determining term weights have been proposed.**
- **Experimentally, *tf.idf* has been found to work well**

- **Given a query *q*, then we score the query against a document *d* using the formula**
- $Score\ (q, d) = \sum(tf.idf_{t,d})$ **where *t* is in $q \cap d$**

Copyright Ellis Horowitz, 2011-2022

Computing TF.IDF -- An Example

Given a document containing 3 terms with given frequencies:

A(3), B(2), C(1)

Assume collection contains 10,000 documents and document frequencies of these 3 terms are:

A(50), B(1300), C(250)

Then:

A: tf = 3/3; idf = log(10000/50) = 5.3; tf.idf = 5.3

B: tf = 2/3; idf = log(10000/1300) = 2.0; tf.idf = 1.3

C: tf = 1/3; idf = log(10000/250) = 3.7; tf.idf = 1.2

Copyright Ellis Horowitz, 2011-2022

Cosine Similarity

- **Distance between vectors d_1 and d_2 is *captured* by the cosine of the angle x between them.**
- **Note – this is a *similarity* measure, not a distance measure**

Copyright Ellis Horowitz, 2011-2022

Similarity Measure

- **A similarity measure is a function that computes the *degree of similarity* between two vectors**
  - **Look back at the previous lecture slides for the definition of similarity**
- **Using a similarity measure between the query and each document has positive aspects:**
  - It is possible to rank the retrieved documents in the order of presumed relevance.
  - It is possible to enforce a certain threshold so that the size of the retrieved set can be controlled.

Copyright Ellis Horowitz, 2011-2022

Normalized Vectors

- A vector can be normalized (given a length of 1) by dividing each of its components by the vector's length
- This maps vectors onto the unit circle:
- Then, $\left|\vec{d}_j\right| = \sqrt{\sum_{i=1}^{n} w_{i,j}^{2}} = 1$
- Longer documents don't get more weight
- For normalized vectors, the cosine is simply the dot product:

$$\cos(\vec{d}_j, \vec{d}_k) = \vec{d}_j \cdot \vec{d}_k$$

Copyright Ellis Horowitz, 2011-2022

Similarity Measure for Document and Query

- **Similarity between vectors for the document d_j and query q can be computed as the vector inner product:**

$$sim(d_j,q) = d_j \bullet q = \sum_{i=1}^{t} w_{ij} \cdot w_{iq}$$

  where w_{ij} is the weight of term i in document j and w_{iq} is the weight of term i in the query
- **For binary vectors, the inner product is the number of matched query terms in the document (size of intersection) (Hamming distance)**
- **For weighted term vectors, it is the sum of the products of the weights of the matched terms.**

Copyright Ellis Horowitz, 2011-2022

Limitations of the Inner Product

- **Favors long documents with a large number of unique terms.**
- **Measures how many terms matched but not how many terms are *not* matched.**

Copyright Ellis Horowitz, 2011-2022

Cosine Similarity Using Inner Product -- Examples

Binary:

	retrieval	database	architecture	computer	text	management	information
– D =	1,	1,	1,	0,	1,	1,	0
– Q =	1,	0 ,	1,	0,	0,	1,	1

Size of vector = size of vocabulary = 7
0 means corresponding term not found in document or query

similarity(D, Q) = 3 **(the inner product)**

Weighted:

$D_1 = 2T_1 + 3T_2 + 5T_3$ $\quad$ $D_2 = 3T_1 + 7T_2 + 1T_3$
$Q = 0T_1 + 0T_2 + 2T_3$

$sim(D_1 , Q) = 2*0 + 3*0 + 5*2 = 10$
$sim(D_2 , Q) = 3*0 + 7*0 + 1*2 = 2$

Copyright Ellis Horowitz, 2011-2022

Cosine Similarity Measure Normalized

- Cosine similarity measures the cosine of the angle between two vectors
- We compute the inner product normalized by the vector lengths

$$CosSim(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}}$$

$D_1 = 2T_1 + 3T_2 + 5T_3$ $\quad$ $CosSim(D_1, Q) = 10 / \sqrt{(4+9+25)(0+0+4)} = 0.81$

$D_2 = 3T_1 + 7T_2 + 1T_3$ $\quad$ $CosSim(D_2, Q) = 2 / \sqrt{(9+49+1)(0+0+4)} = 0.13$

$Q = 0T_1 + 0T_2 + 2T_3$

D_1 is 6 times better than D_2 using cosine similarity but only 5 times better using inner product.

Copyright Ellis Horowitz, 2011-2022

Naïve Implementation

1. Convert all documents in collection D to *tf.idf* weighted vectors, the jth document denoted by d_j, for keywords in vocabulary V
2. Convert each query to a *tf.idf* weighted vector *q*
3. For each d_j in D do
   Compute score $s_j = cosSim(d_j, q)$
4. Sort documents by decreasing score
5. Present top ranked documents to the user

Time complexity: O(|V|·|D|) **Bad for large V & D !**

|V| = 10,000; |D| = 100,000; |V|·|D| = 1,000,000,000

Copyright Ellis Horowitz, 2011-2022

Efficient Cosine Ranking

- Ranking consists of computing the *k* docs in the corpus "nearest" to the query ⇒ *k* largest query-doc cosines.
- To do efficient ranking one must:
  - Compute a single cosine efficiently.
  - Choose the *k* largest cosine values efficiently.

Copyright Ellis Horowitz, 2011-2022

Computing Cosine Scores

cosineScore(q)

1. **float** ***Scores[N] = 0*****;** ***//Scores array for all documents***
2. **float** ***Length[N]*** ***//lengths of all documents***
3. **for each query term** ***t***
4. **do calculate** $w_{t,q}$ **and fetch postings list for** ***t***
5. **for each pair** $(d, tf_{t,d})$ **in postings list**
6. **do** $Scores[d] += w_{t,d} \times w_{t,q}$
7. **Read the array Length**
8. **for each** ***d do***
9. $Scores[d] = Scores[d]/Length[d]$
10. **return Top** ***K*** **components of** ***Scores[]***

weight of query term is *1*;
then,
for each document in the postings list, the term *t* occurs *tf* times;
then we take the dot product of weight of term *t* in document times weight of term *t* in query;

divide scores by length of each document

ranking

in practice we only work with a subset of the documents

Copyright Ellis Horowitz, 2011-2022

Summary of Algorithm for Vector Space Ranking

- Represent the query as a weighted *tf.idf* vector
- represent each document as a weighted *tf.idf* vector
- compute the cosine similarity score for the query vector and each document vector that contains the query term
- Rank documents with respect to the query by score
- Return the top *k* (e.g. *k=10*) to the user

Copyright Ellis Horowitz, 2011-2022

Use Heap for Selecting Top *k*

- Binary tree in which each node's value > values of children
- Takes *2n* operations to construct, then each of *k* log *n* "winners" read off in 2log *n* steps.
- For *n=1M*, *k=100*, this is about 10% of the cost of sorting.

Copyright Ellis Horowitz, 2011-2022

Bottleneck

- Still need to first compute cosines from query to each of *n* docs → several seconds for *n* = 1M
- Can select from only non-zero cosines
  - Need union of postings lists accumulators (<<1M)
- Can further limit to documents with non-zero cosines on rare (high idf) words
- Enforce conjunctive search (a la Google): non-zero cosines on *all* words in query
  - Need min of postings lists sizes accumulators
- But still potentially expensive

Copyright Ellis Horowitz, 2011-2022

A Pre-Processing Strategy

- **<u>Preprocess</u>: Pre-compute, for each term, its *k* nearest docs.**
  - **(Treat each term as a 1-term query)**
  - **lots of preprocessing.**
  - **Result: "preferred list" for each term.**
- **<u>Search</u>:**
  - **For a *t*-term query, take the union of their *t* preferred lists – call this set *S*.**
  - **Compute cosines from the query to only the docs in *S*, and choose top *k*.**

Copyright Ellis Horowitz, 2011-2022

Comments on Vector Space Model

- Simple, mathematically based approach.
- Considers both local (*tf*) and global (*idf*) word occurrence frequencies.
- Provides partial matching and ranked results.
- Tends to work quite well in practice despite obvious weaknesses.
- Allows efficient implementation for large document collections.

Copyright Ellis Horowitz, 2011-2022

Problems with Vector Space Model

- **Missing semantic information (e.g. word sense).**
- **Missing syntactic information (e.g. phrase structure, word order, proximity information).**
- **Assumption of term independence**
- **Lacks the control of a Boolean model (e.g., *requiring* a term to appear in a document).**
  - Given a two-term query "A B", may prefer a document containing A frequently but not B, over a document that contains both A and B, but both less frequently.

Copyright Ellis Horowitz, 2011-2022

Lexicon & Text Normalization

Copyright Ellis Horowitz 2011-2022

Overview of these Slides

- These slides are focused on the issue of how a search engine creates its database of documents.
- In particular how the search engine determines what words go into its inverted index, the **lexicon**
- we begin with a discussion of word frequency and determining the number of unique words (the **vocabulary**)
- then we look at some specific operations for refining the vocabulary
  - Stemming (mentioned earlier)
  - stop words (mentioned earlier)
  - Tokenization (parsing)
  - Capitalization (important or not?)
  - case folding (converting to all uppercase or lowercase)
  - synonyms (thesaurus)
  - similar sounding words

Copyright Ellis Horowitz, 2011-2022

Power Laws and Zipf's Law

- **An equation of the form $y = kx^c$ is called a power law**
  - **k and c are constants**
- **Zipf's law is a power law with $c = -1$**
  - George Zipf noticed the rule while examining word frequencies in text
  - His rule states that the frequency of any word is inversely proportional to its rank in the frequency table
  - Thus the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc
- **On a log-log plot, power laws give a straight line with slope c.**

$$\log(y) = \log(kx^c) = \log k + c\log(x)$$

Copyright Ellis Horowitz, 2011-2022

Some Power Law Examples

- ***Population of U.S. states*:** if we order states in the US by population and let y be the population of the x-th most populous state, then x and y obey Zipf's law approximately.
- ***Book sales at Amazon.com***; let x represent the rank of books by sales and let y be the number of sales of the xth best selling book over some period of time. According to Amazon the best seller sold 1,000,000 copies and the 10th best sold 10,000 copies, and the 100th best selling book sold 100 copies
- ***Node degrees in the web graph***: order all pages by the number of in-links to that page. Let x be the position of a page in this ordering and let y be the number of in-links to the xth page. Then y is a power law function of x

Copyright Ellis Horowitz, 2011-2022

Other Power Law Examples

- **Frequency of accesses to web pages**
  - For example: the access counts on Wikipedia pages
- **Words in the English language**
  - for instance, in Shakespeare's play Hamlet with *c* approximately 0.5
- **Sizes of settlements**
- **Income distributions among individuals**
- **Size of earthquakes**
- **Notes in musical performances**

http://en.wikipedia.org/wiki/Zipf's_law
http://www.cut-the-knot.org/do_you_know/zipfLaw.shtml

Copyright Ellis Horowitz, 2011-2022

Statistical Properties of Text

- **We can ask two questions about words in a set of documents**

1. How is the frequency of different words distributed?
2. How fast does vocabulary size grow with the size of a set of documents?

- **Such factors affect the performance of information retrieval and can be used to select appropriate term weights and other aspects of an IR system.**

Copyright Ellis Horowitz, 2011-2022

Word Frequency

- **A few words are very common.**
  - Two very frequent words (e.g. “the”, “of”) can account for about 10% of word occurrences.
- **Most words are very rare.**
  - Half the words in a corpus appear only once, called *hapax legomena* (Greek for “read only once”)
- **The above phenomenon is called a “heavy tailed” distribution, since most of the probability mass is in the “tail”**
- **See https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists#**
- Here are the top 100 words from Project Gutenberg texts in alphabetical order:
- a · about · after · all · an · and · any · are · as · at · be · been · before · but · by · can · could · did · do · down · first · for · from · good · great · had · has · have · he · her · him · his · I · if · in · into · is · it · its · know · like · little · made · man · may · me · men · more · Mr · much · must · my · no · not · now · of · on · one · only · or · other · our · out · over · said · see · she · should · so · some · such · than · that · the · their · them · then · there · these · they · this · time · to · two · up · upon · us · very · was · we · were · what · when · which · who · will · with · would · you · your

Copyright Ellis Horowitz, 2011-2022

Sample Word Frequency Data (from B. Croft, UMass)

Frequent Word	Number of Occurrences	Percentage of Total
the	7,398,934	5.9
of	3,893,790	3.1
to	3,364,653	2.7
and	3,320,687	2.6
in	2,311,785	1.8
is	1,559,147	1.2
for	1,313,561	1.0
The	1,144,860	0.9
that	1,066,503	0.8
said	1,027,713	0.8

Frequencies from 336,310 documents in the 1GB TREC Volume 3 Corpus
125,720,891 total word occurrences; 508,209 unique words

Copyright Ellis Horowitz 2011-2022

About 0.4% are unique

There are Many Lists of Stop Words

A stop list of 25 semantically non-selective words

a	an	and	are	as	at	be	by	for	from
Has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

- The MOST COMMONLY used list of stop words
- https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html
- Using a stop list significantly reduces the number of postings that a system has to store;
- However stopwords can be important in phrases, e.g. "President of the United States"
- Using compression the added cost of including stopwords is kept in bounds and is generally included in search engines indexes

Copyright Ellis Horowitz 2011-2022

Vocabulary Growth

- **Our second question:** How does the size of the overall vocabulary (number of unique words) grow with the size of the corpus?
- **This determines how the size of the inverted index will scale with the size of the corpus.**
- **In practice a vocabulary is not really upper-bounded due to proper names, typos, etc.**

Copyright Ellis Horowitz, 2011-2022

Heaps' Law

- **If V is the size of the vocabulary and n is the number of words:**

$$V = Kn^{\beta} \quad \text{with constants } K,\ 0 < \beta < 1$$

- **Typical constants:**
  - $K \approx 10 - 100$
  - $\beta \approx 0.4 - 0.6$ **(approx. square-root)**
- So for $n = 100{,}000{,}000$ and $\beta = 0.5$, $K=1$ the vocabulary size is $\approx 10{,}000$
- Heap's law describes the number of distinct words (V) in a set of documents as a function of the document length (n)

The law is named after Harold Stanley Heaps, but was originally discovered by Gustav Herdan

Copyright Ellis Horowitz, 2011-2022

Heaps' Law Data: An Example

Here 40,000,000 words implies a 250,000 vocabulary further implying k = 33 and b=0.5

Copyright Ellis Horowitz 2011-2022

Heap's Law Conclusions

1. the dictionary size continues to increase with more documents in the collection, rather than a maximum vocabulary size being reached
2. the size of the dictionary is quite large for large collections

- 1 and 2 imply that dictionary compression is important to do
- See section 5.2 of our textbook

Copyright Ellis Horowitz, 2011-2022

Lexicon Construction

Copyright Ellis Horowitz 2011-2022

What is a Lexicon?

- **Definition: A database of the vocabulary of a particular domain (or a language)**
- **It is more than a list of words/phrases**
- **Usually it includes some linguistic information**
  - *Morphology* (manag- e/es/ing/ed → manage), (description of the structure of a language's units such as root words, parts of speech, intonations, stresses)
  - *Syntactic patterns* (such as Verb-Object or Subject-Verb-Object)
- **Often some semantic information is included:**
  - *Is-a hierarchy*, e.g.
    - lion is-a primate is-a mammal; corvette is-a Chevrolet is-a General Motors car
  - *Synonyms* (e.g. restaurant, café)
  - Numbers convert to normal form: Four → 4
  - Dates convert to normal form (month/day/year)
  - Alternative names converted to explicit form
    - Mr. Carr, Tyler, Presenter → Tyler Carr

Copyright Ellis Horowitz, 2011-2022

Determining the Characters in a Document

- **Digital documents are typically bytes in a file. The first step of processing is to convert this byte sequence into a linear sequence of characters.**
  - Plain English text in ASCII encoding poses no problem
  - However, multi-byte encoding schemes, such as Unicode are more difficult to identify and process.
  - Determining the correct encoding is often handled by heuristic methods or by using provided document metadata such as:
  - `<html lang="en-US">, <html lang="fr">, <Q lang="he" dir="rtl">`
- **Unicode** (Worldwide Character Standard) is a **character** encoding standard that stores letters and other **characters** by assigning a number for each one.
- Unicode allows for **17** planes, each of **65,536** possible characters (or 'code points'). This gives a total of **1,114,112** possible characters. At present, only about 10% of this space has been allocated
- Unicode may be 8-bit, 16-bit or 32-bit
  - ASCII uses only one byte to represent each character, Unicode up to 4 bytes
- UTF-8 is a popular version of Unicode that uses 8-bit encoding
- There are 3,178 emojis in the Unicode standard as of 10/2019, see
- **https://unicode.org/emoji/charts/full-emoji-list.html**

Copyright Ellis Horowitz, 2011-2022

Determining the Characters

- **Alternatively, the characters may have to be decoded out of some binary representation like Microsoft Word DOC files and/or a compressed format such as zip files**
- **Even for plain text documents, additional decoding may need to be done. In *XML* document's character entities, such as &, need to be decoded to give the correct character**
- **Finally, the textual part of the document may need to be extracted out of other material, e.g. when handling postscript (PS) or PDF files**
- **Solution:** *This problem is usually solved by licensing a software library that handles decoding document formats and character encodings.*

The Apache **Tika™** toolkit detects and extracts metadata and text from over a thousand different file types (such as PPT, XLS, and PDF). All of these file types can be parsed through a single interface, making **Tika** useful for search engine indexing, content analysis, translation, and much more.

Copyright Ellis Horowitz, 2011-2022

More Complications: Format/Language

- Documents being indexed can include items from **many different** languages
  - A single index may contain terms from many languages.
- **Sometimes a single document or its components can contain multiple language formats, e.g.**
  - French email with a German pdf attachment.
  - French email with quoted clauses from an English-language contract
  - French email with Arabic or Hebrew phrases (right-to-left)
- Google recommends **using different URLs for each language version of a page** rather than using cookies or browser settings to adjust the content language on the page.
  - If you use different URLs for different languages, use `hreflang` annotations to help Google search results link to the correct language version of a page
  - `<link rel="alternate" hreflang="en" href=http://www.example.com />`
  - **https://www.youtube.com/watch?v=8ce9jv91beQ&t=23s**

Copyright Ellis Horowitz, 2011-2022

Tokenization

- **Definition**: *tokenization*: The task of chopping a document unit into pieces, called tokens, and possibly throwing away certain characters
- A *token* is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit
- A *term* is a (possibly normalized) type that is included in the lexicon
- A **tokenizer** breaks a stream of text into tokens,
- A **lexer** is basically a **tokenizer**, but it usually attaches extra context to the tokens – e.g. this token is a number, that token is a string literal, this other token is an equality operator.
- **Simple tokenization algorithm**: "chop on whitespace and throw away punctuation characters", but consider these examples:
  - "Mr. O'Neill thinks that the boys' stories about Chile's capital aren't amusing"
  - O'Neill: is the proper token, but according to our rule above it might be treated as one of these five possibilities: neill, oneill, o'neill, o' neill, o neill?
  - aren't: is the proper token, but according to our rule above it might be treated as one of these four possibilities: aren't, arent, are n't, aren t?
- So the simple strategy of splitting on all non-alphanumeric characters can and does fail and one needs a more powerful rule.

Copyright Ellis Horowitz, 2011-2022

Token Normalization

- *Token normalization* is the process of canonicalizing tokens so that matches occur despite superficial differences in the character sequences of the tokens.
- The **standard** way to normalize is to implicitly create *equivalence classes*, which are normally named after one member of the set; this is often called ***stemming***;
  - *stemming* is the process for reducing inflected (or sometimes derived) words to their stem, base or root form – generally a written word form
  - A stemmer reduces the words "fishing", "fished", "fish", and "fisher" to the root word, "fish".
  - Later slides discuss stemming in more detail
- An **alternative** to creating equivalence classes is to maintain relations between un-normalized tokens. This method can be extended to hand-constructed lists of synonyms such as *car* and *automobile*.
- One way is to perform the expansion is during index construction. When the document contains automobile, we index it under car as well (and, usually, also vice-versa)

Copyright Ellis Horowitz, 2011-2022

Handling Unusual Specific Tokens

- A tokenizer should recognize as a single token:
  - email addresses (jblack@mail.yahoo.com),
  - web URLs (http://stuff.big.com/new/specials.html),
  - numeric IP addresses (142.32.48.231),
  - package tracking numbers (1Z9999W99845399981)
  - E.g. aircraft names like B-52,
  - Unusual TV show name such as M*A*S*H
  - phone numbers (800) 234-2333
  - dates (Mar 11, 1983).
- In English, *hyphenation* is used for various purposes ranging from splitting up vowels in words (*co-education*) to joining nouns as names (*Hewlett-Packard*) to a copyediting device to show word grouping (*the hold-him-back-and-drag-him-away maneuver*).
- See our textbook, *Introduction to Information Retrieval* for more details, Section 2.2

Copyright Ellis Horowitz, 2011-2022

Stop Words Can be Critical to Meaning

- **Removal**: for the query: "how to implement the BM25 retrieval formula" it is best to remove stop words "how", "to" and "the" and to focus on "implement BM25 retrieval formula" see https://en.wikipedia.org/wiki/Okapi_BM25
- **Exceptions**: **phrase searches**
  - The meaning of "flights ***to*** London" is likely to be lost if the word ***to*** is stopped out.
  - A search for Vannevar Bush's article "As we may think" will be difficult if the first three words are stopped out, and the system searches simply for documents containing the word think.
  - Some song titles and well known pieces of verse consist entirely of words that are commonly on stop lists (To be or not to be, Let It Be, I don't want to be, ...).
- **Exceptions: sentiment determination**
  - "I told you she was not happy" removing "not" totally changes the sentiment of the sentence

Copyright Ellis Horowitz, 2011-2022

Capitalization/Case-Folding

- *case-folding* is reducing all letters to either all upper case or lower case.
- Often this is a good idea:
  - *automobile* will match *Automobile*
  - *ferrari* will match *Ferrari*

- On the other hand, such case folding can equate words that might better be kept apart. Here are some Proper Nouns that when converted to lower case become something different:
  - *General Motors*,
  - *The Associated Press*
  - *the Fed* vs. *fed*
  - person names such as *Bush*, *Black*
- ***So search engines retain the capitalization as they find it***

Copyright Ellis Horowitz, 2011-2022

Stemming and Lemmatization

- *Stemming* usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving a common base correctly most of the time
- *Lemmatization* refers to the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma*
- **morphology** is the study of **words**, how they are formed, and their relationship to other **words**
  - It analyzes the structure of **words** and parts of **words**, such as stems, root **words**, prefixes, and suffixes.

Form	Suffix	Stem
studies	-es	studi
studying	-ing	study
niñas	-as	niñ
niñez	-ez	niñ

Stemming example; English and Spanish

Form	Morphological information	Lemma
studies	Third person, singular number, present tense of the verb study	study
studying	Gerund of the verb study	study
niñas	Feminine gender, plural number of the noun niño	niño
niñez	Singular number of the noun niñez	niñez

Lemmatization Example

Copyright Ellis Horowitz, 2011-2022

JavaScript Porter Stemmer Online

- **A stemmer written by Martin Porter has become the de-facto standard algorithm used for English stemming**

- The website to the left implements Porter's stemmer algorithm in JavaScript

- Example: "to be or not to be" is not changed at all

Example: Since the department's founding in 1968, our faculty have made pioneering contributions to fundamental and interdisciplinary fields of computing."

Since -> Sinc
department -> depart
founding -> found
pioneering -> pioneer
contributions -> contribut
fundamental -> fundament

The Porter Stemming Algorithm

- The official site is located at http://tartarus.org/~martin/PorterStemmer/
- The site offers versions of the algorithm in multiple languages including C, Perl, Java, JavaScript, PHP

Original paper written in 1979; →

JavaScript version can be found at http://tartarus.org/~martin/PorterStemmer/js.txt

The JavaScript is very easy to follow

Typical Rules in Porter's algorithm

- **Porter's algorithm has five phases of word reductions, applied sequentially**
- **Each phase has a set of conventions for how to select the transformation**
- **For a given set of rules only the rule with longest suffix match is applied; for example**
  - *sses* → *ss*
  - *ies* → *i*
  - *ational* → *ate*
  - *tional* → *tion*

  For Ex. CARESSES maps to CARESS since SSES is the longest match.
- **Other possibilities are suffix stripping algorithms**
  - *If the word ends in 'ed' remove the 'ed'*
  - *If the word ends in 'ing', remove the 'ing'*
  - *If the word ends in 'ly', remove the 'ly'*

Copyright Ellis Horowitz, 2011-2022

Error Metrics

- **Two error measurements**
  - **Over-stemming:** two separate inflected words are stemmed to the same root, but should not have been, a *false positive*
  - **Under-stemming:** where two separate inflected words should be stemmed to the same root, but are not, a *false negative*
- **In the Porter stemmer**
  - Universal, university and universe are stemmed to univers, a case of over-stemming
  - alumnus goes to alumnu, alumni goes to alumni, alumna goes to alumna, but all should stem to the same root, under-stemming

Copyright Ellis Horowitz, 2011-2022

One Last Algorithm Soundex Algorithm

- **Soundex is a phonetic algorithm *for indexing names by their sound* when pronounced in English.**
- **The basic aim is for names with the same pronunciation to be encoded to the same string so that matching can occur despite minor differences in spelling, e.g.**
  - **SMITH and SMYTH** (the second option is pronounced like the first)
- **Soundex is the most widely known of all phonetic algorithms, as it is a standard feature of MySQL, Microsoft SQL Server and Oracle**
- **E.g., *chebyshev* → *tchebycheff***
- **It was developed by Robert Russell and Margaret Oldell in 1918**

Copyright Ellis Horowitz, 2011-2022

Soundex – Typical Algorithm

- Turn every token to be indexed into a 4-character reduced form
- Do the same with query terms
- Build and search an index on the reduced forms
  - (when the query calls for a soundex match)

Here are some good links that discuss the SoundEx Algorithm
https://en.wikipedia.org/wiki/Soundex
https://www.sciencedirect.com/topics/computer-science/soundex-algorithm
http://www.blackwasp.co.uk/soundex.aspx

Copyright Ellis Horowitz, 2011-2022

Soundex – Algorithm

- The algorithm mainly encodes consonants; a vowel will not be encoded unless it is the first letter.
- Every Soundex code consists of a letter and three numbers, e.g. W-252;
  - the letter is the first letter of the surname
  - The numbers are assigned to the remaining letters of the surname according to the Soundex guide
- Washington is coded W-252 (W, 2 for the S, 5 for the N, 2 for the G, and remaining letters disregarded
- Lee is coded L-000 (L the leading letter and since the ee are dropped the zeros are used as padding)

A web page containing versions of Soundex in 36 different programming languages is here
http://rosettacode.org/wiki/Soundex
http://www.archives.gov/research/census/soundex.html

Copyright Ellis Horowitz, 2011-2022

Soundex – Typical Algorithm

1. Retain the first letter of the word.
2. Change all occurrences of the following letters to '0' (zero):
   'A', E', 'I', 'O', 'U', 'H', 'W', 'Y'.
3. Change letters to digits as follows:

- B, F, P, V → 1
- C, G, J, K, Q, S, X, Z → 2
- D,T → 3
- L → 4
- M, N → 5
- R → 6

Copyright Ellis Horowitz, 2011-2022

Soundex Continued

4. **Remove all pairs of consecutive digits.**
5. **Remove all zeros from the resulting string.**
6. **Pad the resulting string with trailing zeros and return the first four positions, which will be of the form <uppercase letter> <digit> <digit> <digit>.**

E.g., *Herman* becomes H655

- **The algorithm is straight forward to code and requires no backtracking or multiple passes over the input word**

Copyright Ellis Horowitz, 2011-2022

Soundex Example

- **Example 1: Zbygniewski has code Z125**
  - starts with Z, consonant in group 1 (the b), one in group 2 (the g), one in group 5 (the n), remainder is ignored
  - if Zbignyefsky is entered, the same code is produced
- **For some words using Soundex does not provide value, e.g.**
- **Example 2: disapont has code D215**
  - a spelling corrector in its database will retrieve all words with code D215 which includes the following 31 words: disband, disbands, disbanded, disbanding, disbandment, disbandments, dispense, dispenses, dispensed, dispensing, dispenser, dispensers, dispensary, dispensaries, dispensable, dispensation, dispensations, deceiving, deceivingly, despondent, despondently, disobeying, disappoint, disappoints, disappointed, disappointing, disappointedly, disappointingly, disappointment, disappointments, disavowing

Copyright Ellis Horowitz, 2011-2022

Postscript: Lucene and Solr

- **Lucene**
  - a high-performance, full-featured text search engine library written entirely in Java.
  - suitable for any application that requires full-text search
  - an open source project available for free download
- **Solr**
  - HTTP-based Search Server
  - Uses XML for configuration
  - Many, many nice features that Lucene users need
    - Faceting, spell checking, highlighting
    - Caching, Replication, Distributed
- Download both from http://lucene.apache.org

Copyright Ellis Horowitz, 2011-2022

Main Lucene Modules

- **Lucene is the underlying software that imports documents and creates an inverted index**
  - **Document Parsing and Analysis**
    - This section is in charge of tokenization
  - **Document Identification**
    - Where the Document ID is created
    - Date of Document is extracted
    - Title of document is extracted
  - **Indexing**
    - Provides access to indexes
    - Maintains indexes

Copyright Ellis Horowitz, 2011-2022

Lucene in a search system

Copyright Ellis Horowitz, 2011-2022

Lucene Tokenizers Specify How the Text in a Field is to be Indexed

- **Tokenizers in Lucene**
  - **WhitespaceTokenizer**
    - divides text at whitespace
  - **SimpleTokenizer**
    - divides text at non-letters
    - converts to lower case
  - **StopTokenizer**
    - SimpleAnalyzer
    - removes stop words
  - **StandardTokenizer (a combination of the above)**
    - good for most European Languages
    - removes stop words
    - convert to lower case
    - https://lucene.apache.org/core/3_0_3/api/core/org/apache/lucene/analysis/standard/StandardTokenizer.html
- **In Lucene you can create you own Tokenizers**

Copyright Ellis Horowitz, 2011-2022

Example 1
Analysis example

- "The quick brown fox jumped over the lazy dog"
- WhitespaceAnalyzer
  - [The] [quick] [brown] [fox] [jumped] [over] [the] [lazy] [dog]
- SimpleAnalyzer
  - [the] [quick] [brown] [fox] [jumped] [over] [the] [lazy] [dog]
- StopAnalyzer
  - [quick] [brown] [fox] [jumped] [over] [lazy] [dog]
- StandardAnalyzer
  - [quick] [brown] [fox] [jumped] [over] [lazy] [dog]

Copyright Ellis Horowitz, 2011-2022

Example 2

- "XY&Z Corporation – xyz@example.com"
- WhitespaceAnalyzer
  - [XY&Z] [Corporation] [-] [xyz@example.com]
- SimpleAnalyzer
  - [xy] [z] [corporation] [xyz] [example] [com]
- StopAnalyzer
  - [xy] [z] [corporation] [xyz] [example] [com]
- StandardAnalyzer
  - [xy&z] [corporation] [xyz@example.com]

Copyright Ellis Horowitz, 2011-2022

TEXT	QUERY
the LexCorp BFG-900 is a printer	Lex corp bfg900 printers

WhitespaceTokenizer

the	LexCorp	BFG-900	is	a	printer	Lex	corp	bfg900	printers

WordDelimiterFilter

the	Lex	Corp LexCorp	BFG	900	is	a	printer	Lex	corp	bfg	900	printers

LowerCaseFilter

the	lex	corp lexcorp	bfg	900	is	a	printer	lex	corp	bfg	900	printers

StopwordFilter

lex	corp lexcorp	bfg	900	printer	lex	corp	bfg	900	printers

StemmerFilter

lex	corp lexcorp	bfg	900	print-	lex	corp	bfg	900	print-

Match

Copyright Ellis Horowitz, 2011-2022

Inverted Indexing

Copyright Ellis Horowitz 2013-2022

Outline

- Definition of an Inverted index
- Examples of Inverted Indices
- Representing an Inverted Index
- Processing a Query on a Linked Inverted Index
- Skip Pointers to Improve Merging
- Phrase Queries
- biwords
- Grammatical Tagging
- N-Grams
- Distributed Indexing

Copyright Ellis Horowitz, 2011-2022

Creating an Inverted Index

- An inverted index is typically composed of a vector containing all distinct words of the text collection in lexicographical order (which is called the **vocabulary**) and for each word in the vocabulary, a list of all documents (and text positions) in which that word occurs
  - This is nothing more than an index that one finds at the back of a book
- Terms in the inverted file index ***may be*** refined:
  - ***Case folding***: converting all letters either to upper case or to lower case
  - ***Stemming/lemmatization***: reducing words to their morphological roots
  - ***Stop words***: removing words that are so common they provide no information

Copyright Ellis Horowitz, 2011-2022

Processing a Query An Example

- The Query
  - Which plays of Shakespeare contain the words *Brutus AND Caesar* but *NOT Calpurnia*?
- One Possible Solution
  - One could grep all of Shakespeare's plays for *Brutus* and *Caesar,* then strip out lines containing *Calpurnia*?
    - Too Slow (for large corpora)
    - Requires lots of space
    - This method doesn't allow for other operations (e.g., find the word *Romans* near *countrymen*) are not feasible with this approach

grep is a command-line utility for searching plain-text data sets for lines that match a regular expression.

Copyright Ellis Horowitz, 2011-2022

Term-Document Incidence Matrix

One way to think about an inverted index is to consider it as a sparse matrix where rows represent terms and columns represent documents

terms ↓ / documents →	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

1 if the play contains word, 0 otherwise

The Query:
Brutus *AND* ***Caesar*** but *NOT* ***Calpurnia***

Copyright Ellis Horowitz, 2011-2022

Incidence Vectors

- So we have a 0/1 vector for each term.
- To answer the previous query: take the vectors for *Brutus, Caesar* and *Calpurnia* (complemented) and do a bitwise *AND*.
- 110100 *AND* 110111 *AND* 101111 = 100100
- So the two plays matching the query are:

  "*Anthony and Cleopatra", "Hamlet"*

Copyright Ellis Horowitz, 2011-2022

Actual Answers to the Query

- **Antony and Cleopatra, Act III, Scene ii**
- ***Agrippa* [Aside to DOMITIUS ENOBARBUS]:**
- Why, Enobarbus,
- When Antony found Julius *Caesar* dead,
- He cried almost to roaring; and he wept
- When at Philippi he found *Brutus* slain.

- **Hamlet, Act III, Scene ii**
- *Lord Polonius:* I did enact Julius *Caesar* I was killed i' the Capitol; *Brutus* killed me.

Copyright Ellis Horowitz, 2011-2022

Term-Document Incident Matrices are Naturally Sparse

- Given 1 million documents and 500,000 terms
- The (term x document) matrix in this case will have size 500K x 1M or half-a-trillion 0's and 1's.
- But it will likely have no more than one billion 1's.
  - So the matrix is extremely sparse, only 0.1% of the elements are 1
- So instead we use a data structure for an inverted index that exploits sparsity and then devise algorithms for query processing

Copyright Ellis Horowitz, 2011-2022

Inverted Index Example

The two parts of an inverted index. The **dictionary** (Index file) is usually kept in memory, with pointers to each **postings list** , which is stored on disk. The dictionary has been sorted alphabetically and the postings list is sorted by document ID

Copyright Ellis Horowitz 2011-2022

Inverted Index Stored In Two Parts

- For each term *T*, we must store a list of all documents that contain *T*.
- Linked lists are generally preferred to arrays, why . . .
  - Dynamic space allocation
  - Insertion of terms into documents easy
  - There is space overhead of pointers, though this is not too serious

Copyright Ellis Horowitz, 2011-2022

Parsing Documents To Create an Inverted Index

- Documents are parsed to extract words and these are saved with the document ID i.e a sequence of (possibly modified token, Document ID) pairs

Doc 1

I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

Doc 2

So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

Term	Doc #
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2
	11

Copyright Ellis Horowitz, 2011-2022

- If the corpus is known in advance, then after all documents have been parsed the inverted file is sorted by terms

Initial capture of terms

Term	Doc #
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Refined list of terms

Term	Doc #
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

- However, on the Web, documents are constantly being added and the terms are constantly increasing

Copyright Ellis Horowitz, 2011-2022

- Multiple term entries in a single document are merged.
- Frequency information is added.

Term	Doc #
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

Term	Doc #	Freq
ambitious	2	1
be	2	1
brutus	1	1
brutus	2	1
capitol	1	1
caesar	1	1
caesar	2	2
did	1	1
enact	1	1
hath	2	1
I	1	2
i'	1	1
it	2	1
julius	1	1
killed	1	2
let	2	1
me	1	1
noble	2	1
so	2	1
the	1	1
the	2	1
told	2	1
you	2	1
was	1	1
was	2	1
with	2	1

Copyright Ellis Horowitz, 2011-2022

e.g. Caesar Occurs in Documents 1 and 2, With Total Frequency 3

- **The file is commonly split into a *Dictionary* and a *Postings* file**

Term	Doc #	Freq
ambitious	2	1
be	2	1
brutus	1	1
brutus	2	1
capitol	1	1
caesar	1	1
caesar	2	2
did	1	1
enact	1	1
hath	2	1
I	1	2
i'	1	1
it	2	1
julius	1	1
killed	1	2
let	2	1
me	1	1
noble	2	1
so	2	1
the	1	1
the	2	1
told	2	1
you	2	1
was	1	1
was	2	1
with	2	1

Term	N docs	Tot Freq
ambitious	1	1
be	1	1
brutus	2	2
capitol	1	1
caesar	2	3
did	1	1
enact	1	1
hath	1	1
I	1	2
i'	1	1
it	1	1
julius	1	1
killed	1	2
let	1	1
me	1	1
noble	1	1
so	1	1
the	2	2
told	1	1
you	1	1
was	2	2
with	1	1

Doc #	Freq
2	1
2	1
1	1
2	1
1	1
1	1
2	2
1	1
1	1
2	1
1	2
1	1
2	1
1	1
1	2
2	1
1	1
2	1
2	1
1	1
2	1
2	1
2	1
1	1
2	1
2	1

Copyright Ellis Horowitz, 2011-2022

Query Processing Across the Postings List

- **Consider processing the query:**

  Brutus AND Caesar

  – Locate *Brutus* in the Dictionary;
    - Retrieve its postings.

  – Locate *Caesar* in the Dictionary;
    - Retrieve its postings.

  – "Merge" the two postings and select the ones in common (postings are document ids):

Copyright Ellis Horowitz, 2011-2022

Basic Merge

- Walk through the two postings simultaneously, in time linear in the total number of postings entries

If the list lengths are *m* and *n*, the merge takes O(*m*+*n*) operations.

Copyright Ellis Horowitz, 2011-2022

Query Optimization

- What is the best order for query processing?
- Consider a query that is an *AND* of *t* terms.
- For each of the *t* terms, get its postings, then *AND* together.

Brutus	2	4	8	16	32	64	128	
Calpurnia	1	2	3	5	8	16	21	34
Caesar	13	16						

Query: ***Brutus*** *AND* ***Calpurnia*** *AND* ***Caesar***

Copyright Ellis Horowitz, 2011-2022

Query Optimization Example

- Process in order of increasing frequency of occurrence:
  - *start with smallest set, then keep cutting further.*

This is why we kept freq in dictionary

Brutus	2	4	8	16	32	64	128	
Calpurnia	1	2	3	5	8	13	21	34
Caesar	13	16						

Execute the query as (***Caesar*** *AND* ***Brutus)*** *AND* ***Calpurnia***.

Copyright Ellis Horowitz, 2011-2022

Algorithm for the intersection of two postings lists p1 and p2

```
INTERSECT( p1, p2)
answer ← ( )
 while p1 /= NIL and p2 /= NIL
 do if docID( p1) = docID( p2)
        then ADD(answer, docID( p1))
             p1 ← next( p1)
             p2 ← next( p2)
        else  if docID( p1) < docID( p2)
                then p1 ← next( p1)
                else  p2 ← next( p2)
 return answer
```

advance pointer on the smaller docID

Copyright Ellis Horowitz, 2011-2022

Algorithm for conjunctive queries that returns the set of documents containing each term in the input list of terms

```
INTERSECT(<t1, . . . , tn>)
terms ← SORTBYINCREASINGFREQUENCY(<t1, . . . , tn>)
result ← postings( first(terms))
terms ← rest(terms)
while terms /= NIL and result /= NIL
do result ← INTERSECT(result, postings( first(terms)))
    terms ← rest(terms)
return result
```

Algorithm Strategy:
intersect each retrieved postings list with the current intermediate result in memory, where we initialize the intermediate result by loading the postings list of the least frequent term

Copyright Ellis Horowitz, 2011-2022

To speed up the merging of postings we use the technique of ***Skip Pointers***

Copyright Ellis Horowitz, 2011-2022

The Technique of Skip Pointers

Augment postings with skip pointers (at indexing time)

- **Why?**
- **<u>To skip postings that will not figure in the search results.</u>**
- **How?**
- **Where do we place skip pointers?**

Query Processing With Skip Pointers

Suppose we've stepped through the lists until we process **8** on each list.

When we get to **16** on the top list, we see that its successor is **32**.

But the skip successor of **8** on the lower list is **31**, so we can skip ahead past the intervening postings 17 and 21

Facts on Skip Pointers

- **Skip pointers are added at indexing time**; they are shortcuts, and they only help for AND queries and they are useful when the corpus is relatively static
- there are two questions that must be answered:
- 1. where should they be placed?
- 2. how do the algorithms change?

- **The Argument:** More skips means shorter skip spans, and that we are more likely to skip. But it also means lots of comparisons to skip pointers, and lots of space storing skip pointers. Fewer skips means few pointer comparisons, but then long skip spans which means that there will be fewer opportunities to skip.

- **The Solution**: A simple heuristic for placing skips, which has been found to work well in practice, is that ***for a postings list of length P, use sqrt{P} evenly-spaced skip pointers.*** This heuristic possibly can be improved upon as it ignores any details of the distribution of query terms. **[Moffat and Zobel 1996]**

.

- See the YouTube video http://www.youtube.com/watch?v=tPsCQOsa7j0 (15 min)

Copyright Ellis Horowitz, 2011-2022

Faster Postings List Intersection via Skip Pointers

```
INTERSECTWITHSKIPS(p1, p2)
answer ← ⟨ ⟩
while p1 ≠ NIL and p2 ≠ NIL
do if docID(p1) = docID(p2)
      then ADD(answer, docID(p1))
           p1 ← next(p1)
           p2 ← next(p2)
      else if docID(p1) < docID(p2)
              then if hasSkip(p1) and (docID(skip(p1)) ≤ docID(p2))
                      then while hasSkip(p1) and (docID(skip(p1)) ≤ docID(p2))
                              do p1 ← skip(p1)
                      else p1 ← next(p1)
              else if hasSkip(p2) and (docID(skip(p2)) ≤ docID(p1))
                      then while hasSkip(p2) and (docID(skip(p2)) ≤ docID(p1))
                              do p2 ← skip(p2)
                      else p2 ← next(p2)
return answer
```

- Skip pointers will only be available for the original postings lists.
- For an intermediate result in a complex query, the call hasSkip(p) will always return false.
- Finally, note that the presence of skip pointers only helps for AND queries, not for OR queries.

Copyright Ellis Horowitz 2011-2022

Phrase Queries

Copyright Ellis Horowitz, 2011-2022

Phrase queries

- We want to answer queries such as *stanford university* – as a phrase
- Many search engines allow users to specify a phrase using double quotes, "Stanford University", which most people find easy to understand
- As many as 10% of web search engine queries are phrase queries and many more are implied phrase queries, e.g. George Bush, Bernie Sanders
- If we only store terms and documents in our inverted index, <*term* : *docs*> then how we can answer a phrase query

- There are two approaches to solving this problem

1. Bi-word indexes (also called 2-grams or 2 shingles)
2. Positional indexes

Copyright Ellis Horowitz, 2011-2022

Using Biword Indexes for Phrase Searching

- Definition: A *bi-word* (or a 2-gram) is a consecutive pair of terms in some text
- To improve phrase searching one approach is to index every bi-word in the text
- For example the text "Friends, Romans, Countrymen" would generate the bi-words
  - *friends romans*
  - *romans countrymen*
- Each of these bi-words is now added to the dictionary as a term

Copyright Ellis Horowitz, 2011-2022

Handling Longer Phrase Queries

- Consequences
  - Bi-words will cause an explosion in the vocabulary database
  - Queries longer than 2 words will have to be broken into bi-word segments
- Example: suppose the query is the 4 word phrase

stanford university palo alto

The query can be broken into the Boolean query on bi-words:

stanford university ***AND*** *university palo* ***AND*** *palo alto*

- Matching the query to terms in the index will work, but may also produce false positives (i.e. occurrences of the bi-words, but not the full 4 word query)
- A Bi-word index that is extended to longer sequences, or even variable length sequences is called a *phrase index*

Copyright Ellis Horowitz, 2011-2022

Part-of-Speech Tagging

- **Many two word phrases have embedded stop words, e.g**
  - the abolition of slavery
  - negotiation of the constitution
- **To salvage the bi-word indexing method on these examples one can use part-of-speech tagging**
  - Part-of-speech taggers classify words as nouns, verbs, etc. – or, in practice, often as finer grained classes like "plural proper noun".
  - "Negotiation of the constitution" is transformed into "N X X N"
- **Many fairly accurate (c. 96% per-tag accuracy) part-of-speech taggers now exist, usually trained by machine learning methods on hand-tagged text**
  - See Manning and Schutze "*Foundations of Statistical Natural Language Processing*"
  - https://nlp.stanford.edu/fsnlp/
  - https://www.issco.unige.ch/en/staff/robert/tatoo/tatoo.html

Copyright Ellis Horowitz, 2011-2022

Alternate Solution - Using Positional Indexes

- Given the limitations of a bi-word index, (i.e. the enormous growth in the vocabulary) the alternate solution is most commonly used, called a **Positional Index**
- Store, for each *term* in the index, entries of the form:

  <term, number of docs containing *term*;

  ***doc1*: position1, position2, position3 … ;**

  ***doc2*: position1, position2 … ;**

  etc.>

- i.e. for each occurrence of the term in a document its position is stored

Copyright Ellis Horowitz, 2011-2022

Positional Index Example

for each term in the vocabulary, we store postings of the form
docID: position1, position2, ...,
where each position is a token index in the document.
Each posting will also usually record the term frequency
Adopting a positional index expands required postings storage significantly, even if we compress position values/offsets

Lots of documents

Lots of occurrences

<***be***: 993427;
1: 7, 18, 33, 72, 86, 231;
2: 3, 149;
4: 17, 191, 291, 430, 434;
5: 363, 367, …>

- the term is "be", typically a stop word
- it occurs in 993,427 documents
- in document 1 "be" occurs at positions: 7, 18, 33, 72, 86 and 231

- **this scheme expands postings storage *substantially (rather than the vocabulary)***

Copyright Ellis Horowitz, 2011-2022

Processing a Phrase Query

- Algorithm for matching a phrase query:
  1. **Extract** inverted index entries for each distinct term: e.g. *to, be, or, not, to, be*
  2. **Merge** their *doc:position* lists to enumerate all positions with "*to be or not to be*".
  3. **Match** those documents that contain the terms in the adjacent positions
  - ***to:***
    - ***2*:1,17,74,222,551; *4*:8,16,190,429,433; *7*:13,23,191; ...**
  - ***be:***
    - ***1*:17,19; *4*:17,191,291,430,434; *5*:14,19,101; ...**
    - **Same general method for proximity searches**
  - In document 4 the word "to" appears in position 16 and the word "be" appears in position 17, so they are adjacent

Copyright Ellis Horowitz, 2011-2022

Algorithm for Proximity Queries with *k* words


```
PositionalIntersect(p1, p2, k)
answer ← ⟨ ⟩
while p1 ≠ NIL and p2 ≠ NIL
do if docID(p1) = docID(p2)
      then l ← ⟨ ⟩
           pp1 ← positions(p1)
           pp2 ← positions(p2)
           while pp1 ≠ NIL
           do while pp2 ≠ NIL
              do if |pos(pp1) − pos(pp2)| ≤ k
                    then Add(l, pos(pp2))
                    else if pos(pp2) > pos(pp1)
                            then break
                 pp2 ← next(pp2)
              while l ≠ ⟨ ⟩ and |l[0] − pos(pp1)| > k
              do Delete(l[0])
              for each ps ∈ l
              do Add(answer, ⟨docID(p1), pos(pp1), ps⟩)
              pp1 ← next(pp1)
           p1 ← next(p1)
           p2 ← next(p2)
      else if docID(p1) < docID(p2)
              then p1 ← next(p1)
              else p2 ← next(p2)
return answer
```

► **Figure 2.12** An algorithm for proximity intersection of postings lists p_1 and p_2. The algorithm finds places where the two terms appear within k words of each other and returns a list of triples giving docID and the term position in p_1 and p_2.

The algorithm finds places where the two terms appear within k words of each other and returns a list of triples giving docID and the term position in p_1 and p_2.

Copyright Ellis Horowitz, 2011-2022

Some High Freqency Noun Phrases from TREC and Patent DataSets

TREC		*Patent*	
Frequency	*Phrase*	*Frequency*	*Phrase*
65824	United States	975362	present invention
61327	article type	191625	u.s. pat
33864	Los Angeles	147352	preferred embodiment
18062	Hong Kong	95097	carbon atoms
17788	North Korea	87903	group consisting
17308	New York	81809	room temperature
15513	San Diego	78458	seq id
15009	Orange county	75850	brief description

The phrases above were identified by POS tagging; The data above shows that common phrases are used more frequently in patent data as patents have a very formal style; many of the TREC phrases are proper nouns, whereas patent phrases are those that occur in all patents

Google Relies Upon *n*-gram Indexes

- Google has investigated the use of n-grams stored in its index, n >= 2;
- N-grams of all lengths form a **Zipf distribution (power law)** with a few common phrases occurring very frequently and a large number occurring with frequency 1
- Google has released the set of n-grams it has determined

Copyright Ellis Horowitz, 2011-2022

Google's N-Gram Database Facts

- Google made available a file of n-grams derived from the web pages it indexed
- http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html
- Statistics for the Google n-gram sample
- Number of tokens 1,024,908,267,229
  (1 trillion, 24 billion, 908 million, . . .)
- 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 four grams 1,313,818,354
- Number of five grams 1,176,470,663

Copyright Ellis Horowitz, 2011-2022

Examples of 3-Gram, 4-Gram Data

Research Blog: All Our N
https://research.googleblog.com/2006/08/all-c
Apps Bookmarks CSCI 571 Home Page Other bookmarks

The following is an example of the 3-gram data contained this corpus:

```
ceramics collectables collectibles 55
ceramics collectables fine 130
ceramics collected by 52
ceramics collectible pottery 50
ceramics collectibles cooking 45
ceramics collection , 144
ceramics collection . 247
ceramics collection </S> 120
ceramics collection and 43
ceramics collection at 52
ceramics collection is 68
ceramics collection of 76
ceramics collection | 59
ceramics collections , 66
ceramics collections . 60
ceramics combined with 46
ceramics come from 69
ceramics comes from 660
ceramics community , 109
ceramics community . 212
ceramics community for 61
```

Research Blog: All Our N
https://research.googleblog.com/2006/08/all-c
Apps Bookmarks CSCI 571 Home Page Other bookmarks

The following is an example of the 4-gram data in this corpus:

```
serve as the incoming 92
serve as the incubator 99
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
serve as the info 42
serve as the informal 102
serve as the information 838
serve as the informational 41
serve as the infrastructure 500
serve as the initial 5331
serve as the initiating 125
serve as the initiation 63
serve as the initiator 81
```

Copyright Ellis Horowitz 2011-2022

The Value of N-Grams

- if we assign a probability to the occurrence of an N-gram or the probability of a word occurring next in a sequence of words, it can be very useful
  - it can help in deciding which N-grams can be chunked together to form single entities (like "San Francisco" chunked together as one word).
  - It can also help make next word predictions. Say you have the partial sentence "Please hand over your". Then it is more likely that the next word is going to be "test" or "assignment" or "paper" than the next word being "school".
  - It can help to make spelling error corrections. For instance, the sentence "drink cofee" could be corrected to "drink coffee" if you knew that the word "coffee" had a high probability of occurrence after the word "drink" and also the overlap of letters between "cofee" and "coffee" is high.

Copyright Ellis Horowitz, 2011-2022

An N-Gram Example

- **Consider the 5 sentences**

1. He said thank you.
2. He said bye as he walked through the door.
3. He went to San Diego.
4. San Diego has nice weather.
5. It is raining in San Francisco.

- **Assume a bi-gram model, i.e. we will find the probability of a word based only on its previous word**
- **Probability (word) = (the number of times the previous word, pw, occurs before the actual word, aw) / (total number of times pw occurs in the corpus)**
- **(count (pw aw)) / (count (pw))**
- **The probability of "you" following "thank", or P (you | thank) =**
- **(No. of times "thank you" occurs) / (no. of time "Thank" occurs)**
- **= 1/1 = 1**

The probability of "Diego" coming after "San" = (No. times "San Diego" occurs) / (No. of times "San" occurs)
= 2/3 = 0.67

Copyright Ellis Horowitz, 2011-2022

N-Grams Used To Find Best and Worst Search Queries

- Lacrosse occurs in 46,579 search terms on Google;
- Overall it produced $843,708 revenue by spending $152,575
- "Lacrosse equipment" was the most profitable, while "girls lacrosse cleats the least

Unigram	Count	Clicks	Cost	Transactions	Revenue	ROAS
lacrosse	46,579	200,727	$152,575	6,614	$843,708	553%

Search Queries Containing "Lacrosse" (Sample)					
Search Query	Impressions	Clicks	Costs	Order	Revenue
lacrosse equipment	13548	637	$493.04	31	$5,306.48
lacrosse gloves	28552	631	$319.14	24	$2,182.20
lacrosse gear	18234	80	$686.99	23	$3,529.44
lacrosse cleats	21965	958	$682.38	18	$2,200.82
box lacrosse bicep pads	275	27	$17.82	10	$521.96
lacrosse sticks	46274	553	$422.03	7	$890.46
lacrosse elbow pads	10299	124	$111.54	6	$303.60
youth lacrosse gloves	2330	105	$73.57	6	$308.53
womens lacrosse cleats	4982	184	$148.76	6	$1,066.30
women's lacrosse cleats	4251	178	$138.19	5	$563.62
lacrosse heads	7334	151	$111.79	5	$550.05
lacrosse bags	8584	157	$80.80	5	$337.56
youth lacrosse gear	1628	122	$109.82	5	$1,369.11
discount lacrosse gear	191	39	$18.57	5	$1,078.36
lacrosse socks	3190	89	$48.09	5	$506.72
lacrosse gear bag	1179	40	$23.13	4	$410.71
discount lacrosse heads	19	6	$3.77	4	$272.78
girls lacrosse cleats	4564	114	$77.77	4	$354.86

Copyright Ellis Horowitz, 2011-2022

Google's N-Gram Viewer

- The **Google N-Gram Viewer** or **Google Books N-Gram Viewer** is an online search engine that charts frequencies of any set of comma-delimited search strings using a yearly count of n-grams found in sources printed between 1500 and 2008 in **Google's** text corpora in English, Chinese (simplified), French, German, Hebrew, Italian, ...
  - https://books.google.com/ngrams/, or for examples see books.google.com/ngrams/info
- The program can search for a single word or a phrase, including misspellings
- The n-grams are matched with the text within the selected corpus, and, if found in 40 or more books, are then plotted on a graph
- The data used for the search are composed of total_counts, 1-grams, 2-grams, 3-grams, 4-grams, and 5-grams files for each language

This shows trends in three n-grams from 1950 to 2000:
"nursery school" (a *2-gram* or *bigram*),
"kindergarten" (a *1-gram* or *unigram*),
and
"child care" (another bigram)

Copyright Ellis Horowitz, 2011-2022

Comparing N-Grams Across Languages

- S. Yang et al, N-gram statistics in English and Chinese: Similarities and differences, ICSC, 2007, Int'l Conf. on semantic computing, 454-460
- http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/pubs/archive/33035.pdf
- The authors above analyzed 200 million randomly sampled English and Chinese Web pages and concluded:

1. The distribution of the unique number of n-grams is similar between English and Chinese, though the Chinese distribution is shifted to larger N
2. The distribution indicates that on average 1.5 Chinese characters correspond to 1 English word
3. While frequency distributions of uni-grams and bi-grams are very different, the frequency distribution for 3-grams and 4-grams are strikingly similar

Copyright Ellis Horowitz, 20

Video Search Engines
YouTube et al

Copyright Ellis Horowitz 2011-2022

Video Search Engines – Quick Summary

- A **video search engine** is a web-based search engine which crawls the web primarily for video content.
  - YouTube is not strictly a video search engine as it does not crawl the web looking for video content
- The ***indexing*** of video content is normally done by acquiring meta-data associated with the video, e.g.
  - Author, title, creation date, duration, coding quality, tags, description
  - Other aspects of video recognition are subtitles (using formats STR or SUB) and transcription (using format TTXT)
- The ***ranking*** of videos under a query is generally done using:
  - Relevance: using metadata and user preferences
  - Ordered by date of upload
  - Ordered by number of views
  - Ordered by duration
  - Ordered by user rating

So indexing and ranking are a lot simpler than for document search engines

Copyright Ellis Horowitz, 2011-2022

Video Search Engines That *Crawl* for Content

- ***Those no longer existing***
  - ***CastTV*** was a Web-wide video search engine that was founded in 2006
    - ***No longer active***
  - ***Munax*** released their first version all-content search engine in 2005 and powers both nationwide and worldwide search engines with video search
    - *http://www.munax.com/* ***no longer active***
  - ***ScienceStage*** is an integrated universal search engine for science-oriented videos. All videos are also semantically matched to millions of research documents from open-access databases.
  - ***No longer active***
- ***A few remain***
  - ***Bing*** does crawl for videos, see https://www.bing.com/videos/
  - ***blinkx*** (renamed as RhythmOne) was launched in 2004 and uses speech recognition and visual analysis to process downloaded video rather than rely on metadata alone
    - *http://www.blinkxtv.com/ now redirects to 360Daily.com*

Copyright Ellis Horowitz, 2011-2022

Video Search Engines That Host

- Largely because of the large file sizes involved, video hosting is highly concentrated on a fairly small number of websites
  - ***vimeo.com***, first to support HD video, focuses on short, arty, films
  - ***vevo.com***, a joint venture of Universal Music Group, Sony Music Entertainment and Warner Music Group
  - ***dailymotion.com***, owned by Vivendi, hosts high quality videos
- Most of these websites which host video allow their videos to be embedded on other websites
- **YouTube.com** has become the defacto site for uploading videos
- It is legal to crawl YouTube, see their Terms of Service, www.youtube.com/static?template=terms
- "3. You are not allowed to access the Service using any automated means (such as robots, botnets or scrapers) except (a) in the case of public search engines, in accordance with YouTube's robots.txt file; or (b) with YouTube's prior written permission;"

Copyright Ellis Horowitz, 2011-2022

Video Search Engines That Stream Entertainment

- **Hulu** is an America subscription video on demand service jointly owned by Walt Disney, 21st Century Fox, Comcast, and Time Warner
  - In December 2017, Disney acquired Fox's partial ownership, giving it a majority stake; other owners include Comcast
- **Netflix** is an American subscription video on demand service, that originally delivered DVDs;
  - They develop their own content as well as offering content from major film distributors
- **Amazon Prime** is an American subscription video on demand service offering television and file shows for rent or purchase
- **Disney+** a recent entry
- There are many others: XtremeHD, Sling TV, Apple TV+, HBO Max, Acorn TV, etc

- **Entertainment**

Copyright Ellis Horowitz, 2011-2022

Some Technologies Supporting Video Content

- **Subtitles:** there are two formats, one for subtitles and one for transcripts
  - There are three main types of video subtitling services:
    1. **open caption:** burned into the video
    2. **closed caption:** can be turned on/off, generally at the bottom of the screen
    3. **SDH (Subtitles for the Deaf and Hard of Hearing):** similar to closed caption, but includes words describing actions or moods
  - **SRT or SUB for subtitles**
    - SRT (. srt) stands for "SubRip Subtitle" file, and it's the most common subtitle/caption file format. It is **a text format**
  - **TTXT for transcripts**
- **Speech Recognition,** used to extract phrases from audio transcripts for better indexing
  - **Gaudi, Google Audio Indexing** uses voice recognition to locate the exact spot where words are spoken
  - https://www.searchenginejournal.com/google-audio-search-will-it-ever-be-possible/397129/
  - **Text Recognition:** uses OCR on video slides to detect words**,**
  - **e.g. TalkMiner System, see https://www.youtube.com/watch?v=7N6I_m9LywM**

Copyright Ellis Horowitz, 2011-2022

YouTube Background

- **YouTube** is an American video hosting website headquartered in San Bruno, California, created by three former PayPal employees: Chad Hurley, Steve Chen, Jawed Karim in February 2005.
- In November 2006, it was bought by Google for US$1.65 billion
- In 2020 Google announced that YouTube generated revenue of $19.8 billion
- The site allows users to upload, view, rate, share, add to favorites, report and comment on videos
- In January 2022, the website was ranked as the second most popular site by Alexa Internet, a web traffic analysis company (now owned by Amazon)
  - See also https://en.wikipedia.org/wiki/List_of_most_popular_websites

For details see Related Articles page, Mar 2020

Copyright Ellis Horowitz, 2011-2022

YouTube as a Search Engine

- YouTube - The 2nd Largest Search Engine (cite:Infographic)
- YouTube processes more than 3 billion searches a month.
- It's bigger than Bing, Yahoo!, Ask and AOL combined!
- http://www.mushroomnetworks.com/infographics/youtube---the-2nd-largest-search-engine-infographic

Copyright Ellis Horowitz, 2011-2022

YouTube Traffic - Some Facts

- As of 2021:
  - 60 hours of video are uploaded every minute, or one hour of video is uploaded to YouTube every second.
  - Over 4 billion videos are viewed a day
  - Over 800 million unique users visit YouTube each month
  - Over 3 billion hours of video are watched each month on YouTube
  - More video is uploaded to YouTube in one month than the 3 major US networks created in 60 years
  - 70% of YouTube traffic comes from outside the US
  - YouTube is localized in 39 countries and across 54 languages
  - It is estimated that YouTube holds 1 sextillion gigabytes of data
  - https://www.quora.com/What-is-the-total-size-storage-capacity-of-YouTube-and-at-what-rate-is-it-increasing-How-is-Google-keeping-up-with-the-increasing-demands-of-Youtube%E2%80%99s-capacity-given-that-thousands-of-videos-are-uploaded-every-day

Copyright Ellis Horowitz, 2011-2022

YouTube Search Engine Issues to Consider

- **Since crawling, indexing and ranking are not big challenges for YouTube, what are the major hurdles**

1. **What video formats are acceptable**
   - For uploading
   - For downloading
2. **How are videos to be displayed on: desktops, iPhones, iPads, Android devices, etc**
3. **How does YouTube distribute videos worldwide**
   - A content distribution network (CDN)
4. **How does YouTube monetize its website?**
   - YouTube's ContentID system
5. **How does YouTube keep users watching**
   - The YouTube Recommendation System

Copyright Ellis Horowitz, 2011-2022

YouTube Channels

- In order to upload a video you must be a registered user
- In addition YouTube offers a special type of account called a ***channel***; channels include
  - thumbnails of videos you've uploaded,
  - members to whom you've subscribed,
  - videos from other members you've picked as favorites,
  - lists of members who are your friends,
  - your subscribers, and
- Biggest YouTube Channels as of 2021

With 1 million subscribers, a YouTuber will make between $300,000 – $2 million
To be in the top 1000 YouTubers you must have ~1.8 million subscribers
As of 09/2020, there are more than 2000 YouTubers with over a million subscribers

https://www.statista.com/statistics/277758/most-popular-youtube-channels-ranked-by-subscribers/

Copyright Ellis Horowitz, 2011-2022

YouTube Gathers Information When Videos are Uploaded

YouTube captures:

Name
Description
Tags

Note: YouTube immediately assigns a URL

Note: YouTube suggests possible thumbnails

Video on How to Upload a Video
https://support.google.com/youtube/answer/57407

Copyright Ellis Horowitz 2011-2022

Uploading to YouTube Second Input Screen

1 of 1 uploaded - YouTube

Ellis

Secure https://www.youtube.com/upload

Apps | CSCI 572 Home Page | CSCI 571 Home Page | CSCI 351 Home Page | Ellis Horowitz' Hom... | Computer Science... | DynDNS -- Host Se... | Other Bookmarks

YouTube Search

PROCESSING DONE

Publish

Click "Publish" to make your video live.

Draft saved.

Basic info | Translations | Advanced settings

Original language

Select language

Translate into (0)

Select language

Upload status:

Upload complete!

Your video will be live at:
https://youtu.be/KX014x5G1Rw

Video / Audio quality:

Your videos will process faster if you encode into a streamable file format. For more information, visit our Help Center.

1 of 2

LennonBirthday

Enter translated title

Description

Enter translated description

Get professional translation

Buy translation (BETA)

Video Manager

+ Add more videos

Uploading to YouTube Third Input Screen

YouTube allows the creator to specify:

License and ownership
Syndication
Caption
Embedding
Age restrictions
Categories
. . .

Business Model: Ads, Ads, Ads

Sample YouTube Search Results for Katy Perry

First result is an Ad

2nd and 4th results are stored at Vevo

3rd and 5th results are links to a Katy Perry channel with 106 videos

To the right is a mix of Katy Perry songs and some "related" artists

Ranking: Ads, Views, Age YouTube Search Results

Begins with an ad

The next 4 results are ordered by the number of views: 420,004, 369,979, 228,004

Subsequent listings are a mixture of highly viewed videos, but older, e.g. Lec 1 MIT has 3 million+ views but is 7 years old

It is not obvious how the ranking was determined

YouTube Advanced Search Ranking Filters

- During a search YouTube provides filters for users to refine their search:
  - UPLOAD DATE
  - TYPE
  - DURATION
  - FEATURES
  - SORT BY

Copyright Ellis Horowitz, 2011-2022

YouTube Ranking Factors

- **YouTube uses the following metrics for ranking search results:**

1. **Meta Data**
   - video titles, descriptions and tags are core ranking factors
   - include links to a website and social profiles
2. **Video Quality**
   - HD ranks higher than low quality videos
3. **Number of views, likes, shares and links**
4. **Subtitles and Closed Captions**
   - captions are crawled by the YouTube search engine and used for ranking

- **What is not known is how YouTube weights the individual factors to make up their final ranking**

Copyright Ellis Horowitz, 2011-2022

YouTube Upload Characteristics

- **YouTube Upload Characteristics**
  - **YouTube** supports 8 video formats for uploading: MOV, MP4 (MPEG4), AVI, WMV, FLV, 3GP, MPEGPS, WebM
  - **Aspect Ratio:** the standard aspect ratios are: 4:3 or 16:9. When the video is uploaded to the site, YouTube will either leave it as-is (for 16:9) or add vertical black bars (for 4:3)
  - **The maximum file size you can upload to YouTube is 128GB.**
  - **By default, you can upload videos that are up to 15 minutes long, though that can be extended**
  - **Many videos have a short life cycle, e.g. a recent tennis match that is soon forgotten, however, there is no time limit for videos to remain on YouTube, unless**
    - You delete the video.
    - You delete your account.
    - You violate copyright or community guidelines

Copyright Ellis Horowitz, 2011-2022

YouTube Videos Run On Multiple Platforms

- **Desktops/laptops**
  - Videos are played in your browser assuming it supports HTML5
  - This avoided the need to use Adobe Flash Player
- **Smartphones**
  - YouTube apps exist for Android and iPhone devices
    - There is no native support for running YouTube videos so a separate app is required
  - For YouTube's videos to run on Apple products YouTube's content had to be transcoded into Apple's preferred video standard, H.264
- **Other Devices**
  - Apple TV, Fire TV, iPod Touch,
  - TiVo, PlayStation, Wii Game consoles,
  - Xbox Live, Roku Players
  - Google Chromecast

Copyright Ellis Horowitz, 2011-2022

YouTube Makes Recommendations to Retain Viewers

- **YouTube Search Results Example for query "computer algorithms"**
- **Assume we choose the first result**

Recommendations are made to maximize watch time

https://www.nbcnews.com/tech/social-media/algorithms-take-over-youtube-s-recommendations-highlight-human-problem-n867596

Copyright Ellis Horowitz, 2011-2022

YouTube Recommendation Algorithm

- Given the query "computer algorithms" followed by a selection, YouTube makes recommendations for subsequent videos
- Recommendations account for 60% of all video clicks

YouTube Recommendation System Uses Graph Properties

- **Association Rule Mining**
  - For each pair of videos v_i v_j compute co-visitation counts, i.e. they count how often they were co-watched; if $c_{i,j}$ is the co-visitation count, then relatedness is defined as

$$r(v_i, v_j) = \frac{c_{ij}}{f(v_i, v_j)}$$

where c_i and c_j are the total occurrence counts across all sessions for videos v_i and v_j. $f(v_i, v_j)$ is a normalization function that takes the global popularity of both the seed video and the candidate video into account; e.g. $f(v_i, v_j) = c_i * c_j$

The set of related videos, R_i for a given seed video v_i is determined by taking the top N candidate videos ranked by their scores $r(v_i, v_j)$

Related videos induce a directed graph over the set of videos, namely:
For each pair of videos (v_i, v_j), there is an edge e_{ij} from v_i to v_j iff v_j is in R_i

For details see: *The YouTube Recommendation System*
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.434.9301&rep=rep1&type=pdf

Copyright Ellis Horowitz, 2011-2022

Media Sites (including YouTube) Move Away from False Information

- **YouTube's recommendation algorithm used to send people to misinformation, e.g. see**
  - https://www.youtube.com/watch?v=FI8tFmBIPak (3 min)
  - https://www.wsj.com/articles/how-youtube-drives-viewers-to-the-internets-darkest-corners-1518020478
- **As a result YouTube changed its recommendation algorithm in 2020, eliminating so-called "fringe" sites**
  - https://www.nytimes.com/2020/11/03/technology/youtube-misinformation-fox-news.html
- **There is a website that tracks the recommendations of Youtube,**
- **https://algotransparency.org/**
- "We used a multi-step program to analyze videos recommended by its algorithm every day"
  - Step 1: We start from a list of 1000+ US channels (these channels are listed below)
  - Step 2: We gather all recommended videos from the last video uploaded by these channels
  - Step 3: We compute which channel was recommended the most from those videos
  - Step 4: We repeat step 2 and 3 until we have gathered recommendations from 2000 channels
  - Step 5: For each video that was observed, we count and display from how many channels it was recommended

Copyright Ellis Horowitz, 2011-2022

Google Search is Biased Towards YouTube Videos

A **video rich snippet** means that when someone searches for something on Google, you can have a small tiny **video** show up next to your result to let the user know that particular result (yours) has a **video** to help

Google weeded out the video competition in Web search by predominantly displaying ***only video-rich snippets*** for YouTube videos back in 2014.

Here is a graph outlining the percentage share of video-rich snippets in Google; 91% are from YouTube

see https://wistia.com/blog/where-did-my-video-snippets-go

e.g. try “tutorial on bitcoin”

Copyright Ellis Horowitz 2011-2022

Content Delivery Networks

- A content distribution network (CDN) consists of a large set of content servers and a means for dynamically selecting servers based on knowledge of the location of the user and possibly the content being requested
- Some sights operate their own CDN, e.g. Google, YouTube
- There are third party companies that offer CDN services such as Akamai, Limelight and Level 3 Communications (now part of Century Link)
- See the Akamai video for 5 minutes (Tom Leighton, start at 0:44-5:00),
- https://www.youtube.com/watch?v=Ni_60cbMydg

Copyright Ellis Horowitz, 2011-2022

YouTube Video Delivery System

- **Two Critical Technology Challenges for YouTube:**
  - *how to identify billions of videos*
  - *How to efficiently deliver the video to the desktop/mobile device*
- **The Solutions:**
- **Identification:** *YouTube assigns a fixed-length, 11 character string, base 64, unique identifier to each video, see*
- https://www.youtube.com/watch?v=gocwRvLhDf8 (5 min)
- **Efficient Delivery:** *YouTube makes use of Google's data centers using them as a content distribution network*
  - https://www.youtube.com/watch?v=6yrijdhvAtI (2 mins)

Copyright Ellis Horowitz, 2011-2022

YouTube (Google's) Content Delivery Datacenters

- A map of Google's data centers, see
- https://www.google.com/about/datacenters/inside/locations/index.html

Figure 4: Geographical distribution of YouTube Video Cache Locations.

Copyright Ellis Horowitz, 2011-2022

Uploading a YouTube Video

1. videos are uploaded from a desktop to a central Data Center
2. the video is then transcoded into multiple formats
3. transcoded copies are sent to the Content Distribution Network

Video transcoding is a technique of converting a video into multiple different formats and resolutions to make it playable across different devices and bandwidths. The technique is also known as *video encoding*. This enables YouTube to stream videos in different resolutions such as *144p, 240p, 360p, 480p, 720p, 1080p & 4K*.

Copyright Ellis Horowitz 2011-2022

YouTube's Content Distribution Network
Downloading a YouTube Video

A local DNS server resolves www.youtube.com and is re-directed to a YouTube server which downloads the page information and a pointer to a YouTube server that can deliver the video, e.g. v23.lscache5.c.youtube.com
The request to v23.lscache5 . .
may be further resolved

4 steps describing the delivery of a YouTube video

http://www-users.cs.umn.edu/~zhzhang/Papers/youtube-tech-report.pdf

Copyright Ellis Horowitz 2011-2022

YouTube Delivery System

- The design of the YouTube video delivery system consists of three components:
  1. a "flat" video id space,
  2. a multi-layered logical server organization consisting of five anycast namespaces (and two unicast namespaces), and
  3. a 3-tiered physical cache hierarchy with (at least) 38 primary locations, 8 secondary and 5 tertiary locations.

Figure 3: YouTube Architectural Design.

https://www-users.cse.umn.edu/~zhang089/Papers/youtube-tech-report.pdf

References to YouTube's CDN

- **There are four research papers that investigated and discussed the YouTube CDN, they are:**
  1. *Vivisecting YouTube: An Active Measurement Study, 2012, cited by Jefay*
  2. *Dissecting Video Server Selection Strategies in the YouTube CDN, 2011, cited by Jefay*
  3. *YouTube Traffic Dynamics and Its InterPlay with a Tier-1 ISP, 2010*
  4. https://www-users.cse.umn.edu/~zhang089/Papers/youtube-tech-report.pdf

- **All of the papers describe a complicated re-direction scheme to find the nearest data center to serve the video; they attempt to minimize Round Trip Time or RTT**
- **For rarely-called-for videos the "*Dissecting" paper* did a study requesting in California a rare video and observed that the first request came from the Netherlands, but future requests were served from California**
-
  - **Conclusion**: videos are constantly being moved around to be closer to the place that is requesting them

Copyright Ellis Horowitz, 2011-2022

Monetizing YouTube

- **YouTube challenges in the early days**
  - YouTube had no way of making money and its infrastructure is very expensive
  - YouTube was being sued by content creators as many of YouTube's videos were uploaded illegally
  - YouTube **solved both problems** at once, by
    - Developing a system for spotting copyrighted content
    - Allowing the copyright owner to decide if he wants to keep the content on the site and let ads appear, splitting the revenue with YouTube or taking the content down
  - Here is a video that describes how YouTubers make money
  - https://www.youtube.com/watch?v=v8F4jrtZtNE
  - (8 min)

Copyright Ellis Horowitz, 2011-2022

ContentID

- **YouTube's solution was to create a fingerprint database of copyrighted content, called Content ID**
- **YouTube solicited cooperation from content owners asking them to submit copies of their content so YouTube could fingerprint them**
  - There are millions of reference files in YouTube's Content ID database.
- **When a new video is uploaded, it is immediately checked against the database, and the video is flagged as a copyright violation if a match is found.**
- **When this occurs, the content owner has the choice of**
  1. blocking the video to make it unviewable,
  2. tracking the viewing statistics of the video, or
  3. adding advertisements to the video

https://arstechnica.com/tech-policy-policy/2014/10/youtube-has-paid-1-billion-to-rights-holders-via-content-id-since-2007/

Copyright Ellis Horowitz, 2011-2022

More Details on ContentID

1. Content ID is based off audio and video samples that rights holders have uploaded to YouTube
2. User uploads a video.
3. YouTube then queues up the video to be processed i.e. it is transcoded into multiple formats including:
   - HTML5, H.264, WebM VP8, HD, non-HD, and others
4. *If the video contains audio*, a hash is then calculated based off a time frequency graph called a spectrogram.
   - Target zones (peak points in the spectrogram) are marked, then the target area between them is also taken and hashed
5. *For the video portion*, a sample section of frames of the video is taken.
   - A hash is created from those sampled frames of the video

- **Note** recently YouTube has introduced a new version of ContentID it calls Copyright Match
- See the following videos for details, (2 min). https://youtu.be/5-2R-IZITZ8

Copyright Ellis Horowitz, 2011-2022

Creating an Acoustic Fingerprint

- **The audio signal is digitized and converted to a spectrogram – a time-frequency graph**
  - The graph below plots three dimensions of audio: frequency versus amplitude versus time
  - A common format is a graph with two dimensions: one axis represents time, and the other axis represents frequency; a third dimension indicating the amplitude of a particular frequency at a particular time is represented by the intensity or color of each point in the image.

frequency

Time axis

How Good is Content ID

- According to stats released by YouTube **99.5 percent** of all copyright issues specifically related to sound recordings are automatically resolved by Content ID
- In addition to music, Content ID also identifies 98% of copyright claims, including those tied to film, TV, gaming
- The actual hashing algorithm used by YouTube remains proprietary, but it has been suggested that YouTube uses finite-state transducers to compute the hash function, e.g. see
- Eugene Weinstein, Pedro J. Moreno; *Music Identification with Weighted Finite-State Transducers*, Proceedings of the International Conference in Acoustics, Speech and Signal Processing (ICASSP), 2007

Copyright Ellis Horowitz, 2011-2022

Will YouTube Ever Run Out of Storage

- The storage you can buy with $100 has grown exponentially — or equivalently, the cost of storing 1GB of videos has decreased exponentially

Kryder's Law
https://en.wikipedia.org/wiki/Mark_Kryder#Kryder%27s_law_projection

Copyright Ellis Horowitz, 2011-2022

Total Capacity of YouTube Storage

https://www.quora.com/What-is-the-total-capacity-of-YouTube-storage

- An answer by Rasty Turek from Quora
- There is roughly 24TB of new videos uploaded daily
- Each video is re-encoded based on pre-selected profiles and each is stored as a separate file
- Here is his computation:

1280x720	mp4	download - 59.01 MB
640x360	mp4	download - 15.34 MB
640x360	webm	download - 19.07 MB
400x240	flv	download - 8.51 MB
320x240	3gp	download - 5.94 MB
176x144	3gp	download - 2.12 MB
4k (no audio)	mp4	download - 297.69 MB

24TB * 4x (for profiles) * 365 days = 35PB/year

So YouTube needs to store roughly 35PB of new data every year.

From multiple sources we know that there is roughly 1Billion videos that have been uploaded to YouTube to date. Assuming each video has on average size of 86MB we can compute their total storage needs as:

86MB * 4 (for profiles) * 1,000,000,000 = 320PB

So it is estimated that YouTube needs to have at least 320PB of storage currently and that the storage needs are growing each year by 35 PetaBytes

Copyright Ellis Horowitz, 2011-2022

Link Analysis Algorithms

Copyright Ellis Horowitz 2011-2022

Overview

- Background on Citation Analysis
- Google's PageRank Algorithm
- Simplified Explanation of PageRank
- Examples of how PageRank algorithm works
- Observations about PageRank algorithm
- Importance of PageRank
- Kleinberg's HITS Algorithm

Copyright Ellis Horowitz, 2011-2022

History of Link Analysis

- **Bibliometrics has been active since at least the 1960's**

- **A definition from Wikipedia:**
- "**Bibliometrics** is a set of methods to quantitatively analyze academic literature. Citation analysis and content analysis are commonly used **bibliometric** methods. While **bibliometric** methods are most often used in the field of library and information science, **bibliometrics** have wide applications in other areas."

- Many research fields use **bibliometric** methods to explore
  - the impact of their field,
  - the impact of a set of researchers, or
  - the impact of a particular paper.

Copyright Ellis Horowitz, 2011-2022

Bibliometrics

- **One common technique of Bibliometrics is *citation analysis***
- **Citation analysis** is the examination of the frequency, patterns, and graphs of citations in articles and books.
- citation analysis can observe links to other works or other researchers.
- **Bibliographic coupling:** two papers that cite many of the same papers
- **Co-citation:** two papers that were cited by many of the same papers
- **Impact factor (of a journal):** frequency with which the average article in a journal has been cited in a particular year or period

http://citeseerx.ist.psu.edu/stats/citations

Top Ten Most Cited Articles in CS Literature

CiteSeer^x is a search engine for academic papers

CiteSeer^x 10M

Most Cited Articles | Most Cited Citations | Most Cited Authors | Venue Impact Ratings

Most Cited Computer Science Citations

This list is generated from documents in the CiteSeer^x database as of March 19, 2015. This list is automatically generated and may contain errors. The list mode and citation counts may differ from those currently in the CiteSeer^x database, since the database is continuously updated.

All Years | 1990 | 1991 | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | | 2015

1. M R Garey, D S Johnson
   Computers and Intractability: A Guide to the Theory of NPCompleteness" W.H. Feeman and 1979
   11468
2. J Sambrook, E F Fritsch, T Maniatis
   Molecular Cloning: A Laboratory Manual, Vol. 1, 2nd edn Nucleic Acids Research, 1989
   10362
3. V Vapnik
   Statistical Learning Theory. 1998
   9898
4. T M Cover, J A Thomas
   Elements of Information Theory Series in Telecommunications, 1991
   9198
5. U K Laemmli
   Cleavage of structural proteins during the assembly of the head of bacteriophage T4. Nature 227:680–685 1970
   9092
6. T H Cormen, C E Leiserson, R L Rivest, C Stein
   Introduction to Algorithms. 1990
   9039
7. A P Dempster, N M Laird, D B Rubin
   Maximum likelihood from incomplete data via the EM algorithm. 1977
   8999
8. D E Goldberg
   Genetic Algorithms in Search, Optimization and Machine Learning, 1989
   8261
9. J Pearl
   Probabilistic Reasoning in Intelligent Systems 1988
   7473
10. C E Shannon, W Weaver
    The Mathematical Theory of Communication 1949
    7077

Copyright Ellis Horowitz 2011-2022

Bibliographic Coupling

- Measure of similarity of documents introduced by Kessler of MIT in 1963.
- The bibliographic coupling of two documents *A* and *B* is the number of documents cited by *both A* and *B*.
- Size of the intersection of their bibliographies.
- Maybe want to normalize by size of bibliographies?

Copyright Ellis Horowitz, 2011-2022

Journal Impact Factor

- Developed by Garfield in 1972 to measure the importance (quality, influence) of scientific journals.
- Measure of how often papers in the journal are cited by other scientists.
- Computed and published annually by the Institute for Scientific Information (ISI).
  - It is now owned by Thomson Reuters
- The *impact factor* of a journal *J* in year *Y* is the average number of citations (from indexed documents published in year *Y*) to a paper published in *J* in year *Y*–1 or *Y*–2.
- Does not account for the quality of the citing article
- https://www.scimagojr.com/journalrank.php?area=1700&category=1702

Copyright Ellis Horowitz, 2011-2022

Top Journals for Computer Science

http://www.guide2research.com/journals/

Over all Computer Science

Top Journals for Computer Science and Electronics

Like 929

Guide2Res... Like Page

Ranking is based on **Impact Factor**. **Vanity press** and poor-quality journals are not listed

All Categories | All Publishers | Search by keyword

Rank	Publisher	Journal Details	Impact Factor
1	IEEE	IEEE Communications Surveys and Tutorials ISSN:1553-877X , Quarterly	9.220
2	IEEE	IEEE Transactions on Fuzzy Systems ISSN:1063-6706 , Bimonthly	6.701
3	IEEE	IEEE Signal Processing Magazine ISSN:1053-5888 , Bimonthly	6.671
4	IEEE	IEEE Transactions on Industrial Electronics ISSN:0278-0046 , Monthly	6.383
5	Mary Ann Liebert	Soft Robotics ISSN:2169-5172 , Quarterly	6.130
6	World Scientific	International Journal of Neural Systems ISSN:0129-0657 , Bimonthly	6.085
7	IEEE	IEEE Transactions on Pattern Analysis and Machine Intelligence ISSN:0162-8828 , Monthly	6.077
8	IEEE	IEEE Transactions on Evolutionary Computation ISSN:1089-778X , Bimonthly	5.908
9	ELSEVIER	Remote Sensing of Environment ISSN:0034-4257 , Monthly	5.881
10	OXFORD UNIVERSITY PRESS	Bioinformatics ISSN:1367-4803 , Semimonthly	5.766

Software Engineering

Top Journals for Computer Science and Electronics

Like 929

Guide2Res... Like Page

Ranking is based on **Impact Factor**. **Vanity press** and poor-quality journals are not listed

Software Engineering | All Publishers | Search by keyword

Rank	Publisher	Journal Details	Impact Factor
90	WILEY	INFORMATION SYSTEMS JOURNAL ISSN:1350-1917 , Bimonthly	2.522
113	IEEE	IEEE Transactions on Reliability ISSN:0018-9529 , Quarterly	2.287
147	Springer	Business and Information Systems Engineering ISSN:1867-0202 , Bimonthly	2.059
185	ELSEVIER	Information Systems ISSN:0306-4379 , Bimonthly	1.832
215	ELSEVIER	Advances in Engineering Software ISSN:0965-9978 , Monthly	1.673
235	IEEE	IEEE Transactions on Dependable and Secure Computing ISSN:1545-5971 , Bimonthly	1.592
237	ELSEVIER	Journal of Computer and System Sciences ISSN:0022-0000 , Bimonthly	1.583
241	ELSEVIER	Information and Software Technology ISSN:0950-5849 , Monthly	1.569
254	IEEE	IEEE Transactions on Software Engineering ISSN:0098-5589 , Monthly	1.516
256	acm Association for Computing Machinery	ACM Transactions on Software Engineering and Methodology ISSN:1049-331X , Quarterly	1.513

Copyright Ellis Horowitz 2011-2022

Citation Graph

The structure of this graph, independent of content, can provide interesting information about the similarity of documents and the structure of information

Note that academic citations nearly always refer to the author's earlier work.

Copyright Ellis Horowitz 2011-2022

Citations vs. Web Links

- **Web links are a bit different than citations:**
  - Many links are navigational
  - Many pages with high in-degree are portals not content providers
  - Not all links are endorsements
  - Company websites normally don't point to their competitors
  - Citations to relevant literature is enforced by peer-review

Copyright Ellis Horowitz, 2011-2022

What is PageRank?

- PageRank is a **web link analysis algorithm** introduced by Google
- PageRank was developed at Stanford University by Google founders **Larry Page and Sergey Brin**
  - The paper describing PageRank was co-authored by Rajeev Motwani and Terry Winograd
- PageRank is a "**vote**", by all the other pages on the Web, about how important a page is.
- A link to a page counts as a vote of support.
- If there's no link there's no support (but it's an abstention from voting rather than a vote against the page).
- PageRank says nothing about the content or size of a page, the language it's written in, or the text used in the anchor of a link!
- Looked at another way, PageRank is a **probability distribution** used to represent the likelihood that a person randomly clicking on links will arrive at any particular page

Copyright Ellis Horowitz, 2011-2022

Page Rank Patented

www.freepatentsonline.com/6285999.pdf

US006285999B1

(12) **United States Patent**
Page

(10) **Patent No.: US 6,285,999 B1**
(45) **Date of Patent: Sep. 4, 2001**

(54) **METHOD FOR NODE RANKING IN A LINKED DATABASE**

(75) Inventor: **Lawrence Page**, Stanford, CA (US)

(73) Assignee: **The Board of Trustees of the Leland Stanford Junior University**, Stanford, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/004,827**

(22) Filed: **Jan. 9, 1998**

Related U.S. Application Data

(60) Provisional application No. 60/035,205, filed on Jan. 10, 1997.

(51) **Int. Cl.**[7] **G06F 17/30**

(52) **U.S. Cl.** **707/5**; 707/7; 707/501

(58) **Field of Search** 707/100, 5, 7, 707/513, 1–3, 10, 104, 501; 345/440; 382/226, 229, 230, 231

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,953,106	*	8/1990	Gansner et al.	345/440
5,450,535	*	9/1995	North	395/140
5,748,954		5/1998	Mauldin	395/610
5,752,241	*	5/1998	Cohen	707/3
5,832,494	*	11/1998	Egger et al.	707/102
5,848,407	*	12/1998	Ishikawa et al.	707/2
6,014,678	*	1/2000	Inoue et al.	707/501

OTHER PUBLICATIONS

S. Jeromy Carriere et al, "Web Query: Searching and Visualizing the Web through Connectivity", Computer Networks and ISDN Systems 29 (1997). pp. 1257–1267.*

Wang et al "Prefetching in Worl Wide Web", IEEE 1996, pp. 28–32.*

Ramer et al "Similarity, Probability and Database Organisation: Extended Abstract", 1996, pp. 272.276.*

Craig Boyle "To link or not to link: An empirical comparison of Hypertext linking strategies". ACM 1992, pp. 221–231.*

L. Katz, "A new status index derived from sociometric analysis," 1953, Psychometricka, vol. 18, pp. 39–43.

C.H. Hubbell, "An input–output approach to clique identification sociometry," 1965, pp. 377–399.

Mizruchi et al., "Techniques for disaggregating centrality scores in social networks," 1996, Sociological Methodology, pp. 26–48.

E. Garfield, "Citation analysis as a tool in journal evaluation," 1972, Science, vol. 178, pp. 471–479.

Pinski et al., "Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics," 1976, Inf. Proc. And Management, vol. 12, pp. 297–312.

N. Geller, "On the citation influence methodology of Pinski and Narin," 1978, Inf. Proc. And Management, vol. 14, pp. 93–95.

P. Doreian, "Measuring the relative standing of disciplinary journals," 1988, Inf. Proc. And Management, vol. 24, pp. 45–56.

(List continued on next page.)

Primary Examiner—Thomas Black
Assistant Examiner—Uyen Le
(74) *Attorney, Agent, or Firm*—Harrity & Snyder L.L.P.

(57) **ABSTRACT**

A method assigns importance ranks to nodes in a linked database, such as any database of documents containing citations, the world wide web or any other hypermedia database. The rank assigned to a document is calculated from the ranks of documents citing it. In addition, the rank of a document is calculated from a constant representing the probability that a browser through the database will randomly jump to the document. The method is particularly useful in enhancing the performance of search engine results for hypermedia databases, such as the world wide web, whose documents have a large variation in quality.

29 Claims, 3 Drawing Sheets

A copy of the front page of the patent of the PageRank algorithm; Larry Page is credited as the inventor; the patent was awarded to Stanford University;
the patent was filed January 1998

The PageRank patent expired in 2017. Google holds a perpetual license to the patent.

Google has never pursued other search engine companies for using the PageRank algorithm

Copyright Ellis Horowitz 2011-2022

Initial PageRank Formulation

$$PR(u) = \sum_{v \in in(u)} \frac{PR(v)}{|out(v)|}$$

➔ "the PageRank value for a page ***u*** is dependent on the PageRank values for each page ***v*** contained in the set ***in(u)*** (the set of all pages linking to page ***u***), divided by the number **|*out(v)*|** of outgoing links from page ***v***"

Copyright Ellis Horowitz, 2011-2022

Steps for Simplified Algorithm

1. **Iteration 0:** Initialize all ranks to be 1/(number of total pages).
2. **Iteration 1:** For each page **u**, update **u**'s rank to be the sum of each incoming page **v**'s rank from the previous iteration, divided by the number total number of links from page **v**.

Copyright Ellis Horowitz, 2011-2022

The Simplified PageRank Algorithm

Example 1

	Iteration 0	Iteration 1
P_1	1/5	1/20
P_2	1/5	5/20
P_3	1/5	1/10
P_4	1/5	5/20
P_5	1/5	7/20

1. <u>Iteration 0:</u> Initialize all pages to have rank ⅕.
2. <u>Iteration 1:</u>
3. **P_1:** has 1 link from P_3, and P_3 has 4 outbound links, so we take the rank of P_3 from iteration 0 and divide it by 4, which results in rank (⅕)/4 = 1/20 for P1

$$PR(P_1) = (\tfrac{1}{5})/4 = 1/20$$

4. **P_2:** has 2 links from P_1 and P_3, P_1 has 1 outbound link and P_3 has 4 outbound links, so we take (the rank of P_1 from iteration 0 and divide it by 1) and add that to (the rank of P_3 from iteration 0 and divided that by 4) to get ⅕ + 1/20 = 5/20 for P_2

$$PR(P_2) = \tfrac{1}{5} + (\tfrac{1}{5})/4 = 5/20$$

Copyright Ellis Horowitz, 2011-2022

Example 1: After 2 iterations

	Iteration 0	Iteration 1	Iteration 2	Final Ranking
P_1	1/5	1/20	1/40	5
P_2	1/5	5/20	3/40	4
P_3	1/5	1/10	5/40	3
P_4	1/5	5/20	15/40	2
P_5	1/5	7/20	16/40	1

$PR(P_5) = \frac{1}{5} + \frac{1}{5} * \frac{1}{4} + \frac{1}{5} * \frac{1}{2} = 7/20$

Copyright Ellis Horowitz, 2011-2022

Example 2

- Say we have four pages: **A**, **B**, **C** and **D**.
- Links from a page to itself are ignored (there aren't any in this example).
- Multiple links from one page to another are treated as a single link.
- In this example, every page would start out with a rank of 0.25

Copyright Ellis Horowitz, 2011-2022

Example 2

- Since B, C, and D all have outbound links to A, the Pagerank of A will be **0.75** upon the first iteration
  - (B with rank of 0.25) + (C with rank of 0.25) + (D with rank of 0.25) would transfer all of those ranks to A
- But wait! What about ranks of pages B,C, and D? Because B, C, and D have no incoming edges and they give all their rank to A, they will all end up with a rank of 0. This doesn't add up to 1 . . .
- ***So the simplified algorithm needs to be able to handle border cases, so we must generalize the PageRank algorithm!***

Copyright Ellis Horowitz, 2011-2022

Complete PageRank Algorithm

- **Quoting from the original Google paper, PageRank is defined like this:**

  "We assume page A has pages T1...Tn which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. We usually set d to 0.85. There are more details about d in the next section. Also C(A) is defined as the number of links going out of page A."*

  The PageRank of a page A is given as follows:

$$PR(A) = (1-d) + d\,(PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn))$$

- ***Note:***
  - ***That the PageRanks form a probability distribution over web pages, so the sum of all web pages' PageRanks will be one.***

- ****The Anatomy of a Large-Scale Hypertextual Web Search Engine by Brin and Page, http://infolab.stanford.edu/pub/papers/google.pdf***

Copyright Ellis Horowitz, 2011-2022

Explanation

$$PR(A) = (1-d) + d\,(PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn))$$

- ***PR(A) is PageRank of Page A (one we want to calculate)***
- ***PR(T1) is the PageRank of Site T1 pointing to page A***
- ***C(T1) is the number of outgoing links of page T1***
- ***PR(Tn)/C(Tn) : If page A has a backlink from page "Tn" the share of the vote page A will get is "PR(Tn)/C(Tn)"***
- **d(…) : All these fractions of votes are added together but, to stop the other pages having too much influence, this total vote is "damped down" by multiplying it by the factor "d", (0.85)**
- **(1-d) : Since "*sum of all web pages' PageRanks will be one*". Even if the d(…) is 0 then the page will get a small PR of 0.15**

Copyright Ellis Horowitz, 2011-2022

How PageRank is Calculated

- **PR of each page depends on PR of other pages which are pointing to it. But we don't know PR of a given page until the PR of other pages is calculated and so on…**
- **From the Google paper:**

  PageRank or PR(A) can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web
- **What this means is that we can calculate the PR without knowing the final PR of other pages**
  - **Recurrence Relation: an equation that recursively defines a sequence of values once one or more initial terms are given;**
- **We calculate PR iteratively a number of times until we start converging to the same value.**

Copyright Ellis Horowitz, 2011-2022

How Fast Does the PageRank Algorithm Converge

- Early experiments on Google used 322 million links
- PR (322 Million Links): 52 iterations
- PR (161 Million Links): 45 iterations
- Number of iterations required for convergence is empirically (but not formally derived) *O(log n)* (where *n* is the number of links)
- Hence the calculation is quite efficient

Copyright Ellis Horowitz, 2011-2022

Computing PageRank by Iteration Example

- Consider 2 pages: Page A and Page B pointing to each other.

- C(A) = 1, number of outgoing links of A
- C(B) = 1, number of outgoing links of B
- What should be the initial value of P(A) or P(B) ?

Copyright Ellis Horowitz, 2011-2022

Guess 1:

- **Suppose the initial values are :**
  - **P(A) = 1 and P(B) = 1 and d= 0.85; then**

    PR(A)= (1 – d) + d*(PR(B)/1)

    PR(B)= (1 – d) + d*(PR(A)/1)

i.e.

- **PR(A)= 0.15 + 0.85 * 1**
  = 1
- **PR(B)= 0.15 + 0.85 * 1**
  = 1
- **In one iteration we are done**
- **Let's try another set of initial values.**

Copyright Ellis Horowitz, 2011-2022

Guess 2 With 3 Iterations

- **Initial Values : P(A) = 0, P(B) = 0 and d= 0.85**

PR(A)= (1 – d) + d(PR(B)/1)
PR(B)= (1 – d) + d(PR(A)/1)

- **PR(A)= 0.15 + 0.85 * 0 = 0.15**
  PR(B)= 0.15 + 0.85 * 0.15 = 0.2775

Iterating again we get:

- **PR(A)= 0.15 + 0.85 * 0.2775 = 0.385875**
  PR(B)= 0.15 + 0.85 * 0.385875 = 0.47799375

And iterating again

- **PR(A)= 0.15 + 0.85 * 0.47799375 = 0.5562946875**
  PR(B)= 0.15 + 0.85 * 0.5562946875 = 0.62285048437

Copyright Ellis Horowitz, 2011-2022

Guess 2: Continued…

- After 20 iterations…
- PR(A) = 0.99
- PR(B) = 0.99
- Both approaching to 1.

	A	B	C	D
1	C(A)	1		
2	C(B)	1		
3				
4	Iterations	PR(A)	PR(B)	
5				
6	0	0	0	
7	1	0.15	0.2775	
8	2	0.385875	0.47799375	
9	3	0.556294688	0.622850484	
10	4	0.679422912	0.727509475	
11	5	0.768383054	0.803125596	
12	6	0.832656756	0.857758243	
13	7	0.879094506	0.89723033	
14	8	0.912645781	0.925748914	
15	9	0.936886577	0.94635359	
16	10	0.954400552	0.961240469	
17	11	0.967054399	0.971996239	
18	12	0.976196803	0.979767283	
19	13	0.98280219	0.985381862	
20	14	0.987574582	0.989438395	
21	15	0.991022636	0.99236924	
22	16	0.993513854	0.994486776	
23	17	0.99531376	0.996016696	
24	18	0.996614191	0.997122063	
25	19	0.997553753	0.99792069	
26	20	0.998232587	0.998497699	
27				

Guess 3:

- **Initial Values : P(A) = 40 and P(B) = 40**
  d= 0.85

PR(A)= (1 – d) + d(PR(B)/1)

PR(B)= (1 – d) + d(PR(A)/1)

- **Notice decreasing value of PR**
- **The page rank is again approaching to 1.**
- **So it doesn't matter where you start your guess, once the PageRank calculations have settled down, the "*normalized probability distribution*" (the average PageRank for these two pages) will be 1.0**

	A	B	C	D
1	C(A)	1		
2	C(B)	1		
3				
4	Iterations	PR(A)	PR(B)	
5				
6	0	0	40	
7	1	34.15	29.1775	
8	2	24.950875	21.35824375	
9	3	18.30450719	15.70883111	
10	4	13.50250644	11.62713048	
11	5	10.03306091	8.678101769	
12	6	7.526386504	6.547428528	
13	7	5.715314249	5.008017112	
14	8	4.406814545	3.895792363	
15	9	3.461423509	3.092209982	
16	10	2.778378485	2.511621712	
17	11	2.284878455	2.092146687	
18	12	1.928324684	1.789075981	
19	13	1.670714584	1.570107397	
20	14	1.484591287	1.411902594	
21	15	1.350117205	1.297599624	
22	16	1.252959681	1.215015728	
23	17	1.182763369	1.155348864	
24	18	1.132046534	1.112239554	
25	19	1.095403621	1.081093078	
26	20	1.068929116	1.058589749	
27				

PageRank Convergence?

- Some observations about the damping factor
- The damping factor value and its effect:
  - For certain graphs the simple update rule can cause pagerank to accumulate and get stuck in certain parts of the graph
    - E.g. if a page has no outgoing links to other pages, it is called a sink
    - The simplified pagerank algorithm can get stuck in sinks
  - This is fixed by having each node on every round
    - Give a *d* fraction of its pagerank to its neighbors
    - Give a *(1-d)* fraction of its pagerank to everyone in the graph
    - As a result, pages with no incoming links will get some pagerank
  - If too high, more iterations are required
  - If too low, you get repeated over-shoot,
  - Both above and below the average – the numbers just swing like pendulum and never settle down.

Copyright Ellis Horowitz, 2011-2022

Matrix Formulation for Computing PageRank

- Suppose we define a matrix M to have one row and one column for each web page
- Suppose page *j* has *n* outlinks
  - If *j* points to *i*, then $M_{ij}=1/n$
  - Else $M_{ij}=0$
- M is a column stochastic matrix
  - i.e. its columns sum to 1
- Suppose *r* is a vector with one entry per web page
  - r_i is the importance score of page *i*
  - Call it the rank vector

Copyright Ellis Horowitz, 2011-2022

Eigenvalue and Eigenvector

- **Eigenvalues and Eigenvectors are properties of a matrix**
- **In general, a matrix acts on a vector by changing both its magnitude and direction**
- **However, a matrix may act on certain vectors by changing only their magnitude, and leaving their direction unchanged – *Eigenvector***
- **A matrix acts on an eigenvector by multiplying its magnitude by a factor called the *Eigenvalue***

Given a linear transformation A, a non-zero vector x is defined to be an eigenvector of the transformation if it satisfies the eigenvalue equation

$$A\mathbf{x} = \lambda\mathbf{x}$$

In this situation, the scalar λ is called an *eigenvalue* of *A* corresponding to the eigenvector x

Copyright Ellis Horowitz, 2011-2022

Flow Equation in Matrix Form

Suppose page i links to 3 pages, $d_i = 3$, including j

d_i is the degree of i or # of out edges

The flow equations can be written

$$r = Mr$$

So the rank vector is an eigenvector of the stochastic web matrix

In fact, its first or principal eigenvector, with corresponding eigenvalue 1

Copyright Ellis Horowitz 2011-2022

Power Iteration method

- Simple iterative scheme (aka relaxation)
- Suppose there are N web pages
- Initialize: $r^0 = [1/N,\ldots,1/N]^T$
- Iterate: $r^{k+1} = Mr^k$
- Stop when $|r^{k+1} - r^k|_1 < \varepsilon$
  - $|x|_1 = \sum_{1 \cdot i \cdot N} |x_i|$ is the L_1 norm
  - Can use any other vector norm e.g., Euclidean

Copyright Ellis Horowitz, 2011-2022

Examples of Relative PageRanks 1:

- **Observations:**
  - **Every page has at least a PR of 0.15 to start out**
  - **Page D has no votes but still it has a PR of 0.15**
  - **It is believed that Google undergoes a post-spidering phase whereby any pages that have no incoming links at all are ignored wrt PageRank**

  - **Examples on the following pages are taken from http://www.sirgroane.net/google-page-rank/**

Example 2: Simple hierarchy with Some Outgoing Links

- **Observations:**
  - **Home has the most PR**
  - **But average PR is 0.378**
    - **"External site" pages are not voting for anyone.**
  - **Links within your own site can have a significant effect on PageRank.**

Example 3: Linking External Sites Back into our Home Page

Average PR: 1.000

- **Observations:**
  - **Comparing this example with the previous one, now the Pagerank of the Home Page has increased significantly.**
  - **Moreover, the average PageRank is better because of all the external sites linking back to the home page.**

Example 4: Simple Hierarchy

- **Observations:**
  - **Home Page has PageRank of 2.5 times the page rank of its child pages.**
  - **A hierarchy structure concentrates votes and PageRank into one page.**

Example 5: Hierarchical – But with One Link In and One Out

- **Observations:**
  - **The PageRank of Home page has increased from 1.92 (Previous Example) to 3.31**
  - **Site A contributed 0.85 PR to Home page and the raised PageRank in the "About", "Product" and "More" pages has had a lovely "feedback" effect, pushing up the home page's PageRank even further!**
- **A well structured site will amplify the effect of any contributed PR.**

Example 6: Looping

- **Observations:**
  - **All the pages have the same number of incoming links, i.e. all pages are of equal importance to each other.**
  - **Each page has PR of 1.0**
  - **Average PR is 1.0**

Example 7: Looping – But with a Link in and a Link Out

- **Observations:**
  - **PageRank of our home page has gone up a little, but what's happened to the "More" page? Its PageRank has gone down**
  - **Now the Pagerank value of the external Site B is equal to the "More" page.**
  - **The vote of the "Product" page has been split evenly between "More" page and the external site B.**
  - **This is good for Site B for otherwise its PageRank would be 0.15**

Example 8: Extensive Interlinking or Fully Meshed

- **Observations:**
  - **All pages are of equal importance to each other so the resulting PageRank is no different than what was gotten in Example 6**

Example 9: Fully Meshed – But with One Vote in and One Vote Out

- **Observations:**
  - **"Product" page has kept three quarters of its vote within our site unlike example 9 where it was giving away fully half of it's vote to the external site!**
  - **Keeping just this small extra fraction of the vote within our site has had a very nice effect on the Home Page too.**

Example 10: Previous … Next … Documentation Page Layout

- **The first page of the document has a higher PR than the Home Page! This is because page B is getting all the vote from page A, but page A is only getting fractions of pages B, C and D.**
- **In order to give users of your site a good experience, you may have to take a hit against your PR**

Example 11: Getting Higher PR the Wrong Way!

Average PR: 1.000

- **Observations:**
  - **1000 incoming links and only one outgoing link**
  - **It doesn't matter how many pages you have in your site, your average PR will always be 1.0 at best.**
  - **But a hierarchical layout can strongly concentrate votes, and therefore the PR, into the home page!**
  - **This is a technique used by some disreputable sites**

A link farm is set of web pages created with the sole aim of linking to a target page, in an attempt to improve that page's search engine ranking.

Some Suggestions Based on What We Have Seen in Examples.

- **Suggestions for improving your page rank**
  - Increasing the internal links in your site can minimize the damage to your PR when you give away votes by linking to external sites.
  - If a particular page is highly important – use a hierarchical structure with the important page at the "top".
  - Where a group of pages may contain outward links – increase the number of internal links to retain as much PR as possible.
  - Where a group of pages do not contain outward links – the number of internal links in the site has no effect on the site's average PR. You might as well use a link structure that gives the user the best navigational experience.
  - **Use Site Maps**: Linking to a site map on each page increases the number of internal links in the site, spreading the PR out and protecting you against your vote "donations" to other sites

Copyright Ellis Horowitz, 2011-2022

Importance of PageRank

- **PageRank is just one factor Google uses to determine a page's relevance. It assumes that people will link to your page only if they think the page is good. But this is not always true.**
- **Content is still the king!!!**
  - Anchor, body, title tags etc. still are very important for search engines
- **From Chris Ridings' Paper, "PageRank Uncovered" (http://www.voelspriet2.nl/PageRank.pdf):**
  - You must do enough "on the page" work and/or anchor text work to get into that subset of top pages for your chosen key phrase, otherwise your high PageRank will be completely in vain.
- **PageRank is a multiplier factor.**
  - If Google wants to penalize any page they can set the PageRank equal to a small number, even 0. As a result it will be pushed down on the search results page.

Copyright Ellis Horowitz, 2011-2022

Random Walk Interpretation

- Imagine a random web surfer
  - 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 *p(t)* be a vector whose i^{th} component is the probability that the surfer is at page *i* at time *t*
  - *p(t)* is a probability distribution on pages

Copyright Ellis Horowitz, 2011-2022

Spider traps

- A group of pages is a spider trap if there are no links from within the group to outside the group
  - Random surfer gets trapped
- Spider traps violate the conditions needed for the random walk theorem

Copyright Ellis Horowitz, 2011-2022

References

- **The original PageRank paper by Google's founders Sergey Brin and Lawrence Page –**
  - http://www-db.stanford.edu/~backrub/google.html
- **PageRank explained correctly:**
  - http://www.iprcom.com/papers/pagerank/
- **Chris Ridings' "PageRank Explained" paper which, as of April 2002**
  - http://web.archive.org/web/*/http://www.goodlookingcooking.co.uk/PageRank.pdf
- **Tool to calculate PageRank**
  - www.markhorrell.com/seo/pagerank.asp
- ***The PageRank Citation Ranking: Bringing Order to the Web*; by L. Page, S. Brin, R. Motwani, T. Winograd, January, 1998,** http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf
- ***The Google PageRank Algorithm and How it Works*, by Ian Rogers, April 2002,** http://www.sirgroane.net/google-page-rank/
- ***PageRank* on Wikipedia,** http://en.wikipedia.org/wiki/PageRank

Copyright Ellis Horowitz, 2011-2022

Another Link Analysis Algorithm – HITS by Kleinberg

- HITS stands for "Hyperlink-Induced Topic Search"
- HITS is a link analysis algorithm developed by Jon Kleinberg at Cornell
- HITS preceded PageRank
- The algorithm is based on the observations that
  - Some pages known as ***hubs*** serve as large directories of information on a given topic
  - Some pages known as ***authorities*** serve as the page(s) that best describe the information on a given topic
- The algorithm begins by retrieving a set of results to a search query
- The HITS algorithm is performed ONLY on the resulting set
- The algorithm goes through a set of iterations where an ***authority value*** is computed as the sum of the scaled hub values that point to that page; and a ***hub value*** is the sum of the scaled authority values of the pages it points to

Authorities

- ***Authorities*** are pages that are recognized as providing significant, trustworthy, and useful information on a topic.
- ***In-degree*** (number of pointers to a page) is one simple measure of authority.
- **However in-degree treats all links as equal.**
- **Should links from pages that are themselves authoritative count more?**

Copyright Ellis Horowitz, 2011-2022

Hubs

- ***Hubs*** are index pages that provide lots of useful links to relevant content pages (topic authorities).
- **Ex: pages are included in the course home page**

Copyright Ellis Horowitz, 2011-2022

Hubs and Authorities

- **Together they tend to form a bipartite graph:**

A bipartite graph is one whose vertices can be divided into two disjoint sets, U and V, such that every edge connects a vertex in U to one in V

Copyright Ellis Horowitz, 2011-2022

The General Idea

- **Authorities- pages that are relevant and are linked to by many other pages**
- **Hubs – pages that have links to multiple relevant authorities**

out-degree: # nodes pointed by

in-degree: # nodes pointing to

Hubs

Authorities

HITS Algorithm

- **Computes hubs and authorities for a particular topic specified by a normal query.**
  - Thus query dependent ranking
- **First determine a set of relevant pages for the query called the *base* set *S*.**
- **Analyze the link structure of the web subgraph defined by *S* to find authority and hub pages in this set.**

Copyright Ellis Horowitz, 2011-2022

Constructing a Base Subgraph

- For a specific query *Q*, let the set of documents returned by a standard search engine be called the *root* set *R*.
- Initialize *S* to *R*.
- Add to *S* all pages pointed to by any page in *R*.
- Add to *S* all pages that point to any page in *R*.

Copyright Ellis Horowitz, 2011-2022

Base Limitations

- **To limit computational expense:**
  - Limit number of root pages to the top 200 pages retrieved for the query.
  - Limit number of "back-pointer" pages to a random set of at most 50 pages returned by a "reverse link" query.
- **To eliminate purely navigational links:**
  - Eliminate links between two pages on the same host.
- **To eliminate "non-authority-conveying" links:**
  - Allow only m ($m \cong 4$–8) pages from a given host as pointers to any individual page.

Copyright Ellis Horowitz, 2011-2022

More Algorithm Detail

- The algorithm performs a series of iterations, each consisting of two basic steps:

1. **Authority Update**: Update each node's *Authority score* to be equal to the sum of the *Hub Scores* of each node that points to it. That is, a node is given a high authority score by being linked to by pages that are recognized as Hubs for information.
2. **Hub Update**: Update each node's *Hub Score* to be equal to the sum of the *Authority Scores* of each node that it points to. That is, a node is given a high hub score by linking to nodes that are considered to be authorities on the subject.

- The Hub score and Authority score for a node is calculated with the following algorithm:
  1. Start with each node having a hub score and authority score of 1.
  2. Run the Authority Update Rule
  3. Run the Hub Update Rule
  4. Normalize the values by dividing each Hub score by the sum of the squares of all Hub scores, and dividing each Authority score by the sum of the squares of all Authority scores.
  5. Repeat from the second step as necessary.

Copyright Ellis Horowitz, 2011-2022

Iterative Algorithm

- **Use an iterative algorithm to slowly converge on a mutually reinforcing set of hubs and authorities.**
- **Maintain for each page $p \in S$:**
  - Authority score: a_p (vector a)
  - Hub score: h_p (vector h)
- **Initialize all $a_p = h_p = 1$**
- **Maintain normalized scores:**

$$\sum_{p \in S} (a_p)^2 = 1 \qquad \sum_{p \in S} (h_p)^2 = 1$$

Copyright Ellis Horowitz, 2011-2022

Convergence

- Algorithm converges to a *fix-point* if iterated indefinitely.
- Define A to be the adjacency matrix for the subgraph defined by S.
  - $A_{ij} = 1$ for $i \in S, j \in S$ iff $i \rightarrow j$
- Authority vector, a, converges to the principal eigenvector of $A^T A$
- Hub vector, h, converges to the principal eigenvector of AA^T
- In practice, 20 iterations produces fairly stable results.

Copyright Ellis Horowitz, 2011-2022

Final Observations

- This algorithm is executed at query time, not at indexing time
  - So there is a penalty on performance
- It computes two scores for each page, a hub score and an authority score
- The method was patented as US 6,112,202 entitled "*Method and System for Identifying Authoritative Information Resources in an Environment with Content-based Links Between Information Resources*"

Copyright Ellis Horowitz, 2011-2022

www.freepatentsonline.com/6112202.pdf

USC Viterbi School o | Main Page - Comput | Faculty Resources, C | Computer Science D | Google | Gmail - Inbox (3)

US006112202A

United States Patent [19]

Kleinberg

[11] **Patent Number: 6,112,202**

[45] **Date of Patent: Aug. 29, 2000**

[54] **METHOD AND SYSTEM FOR IDENTIFYING AUTHORITATIVE INFORMATION RESOURCES IN AN ENVIRONMENT WITH CONTENT-BASED LINKS BETWEEN INFORMATION RESOURCES**

[75] Inventor: **Jon Michael Kleinberg**, Los Gatos, Calif.

[73] Assignee: **International Business Machines Corporation**, Armonk, N.Y.

[21] Appl. No.: **08/813,749**

[22] Filed: **Mar. 7, 1997**

[51] **Int. Cl.**[7] **G06F 17/30**

[52] **U.S. Cl.** **707/5**; 707/9; 707/101

[58] **Field of Search** 707/1, 2, 4, 5, 707/10, 100, 101, 102, 501

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,257,185	10/1993	Farley et al.	707/100
5,446,891	8/1995	Kaplan et al.	707/2
5,778,363	7/1998	Light	707/5
5,826,031	10/1998	Nielsen	395/200.63
5,835,905	11/1998	Pirolli et al.	707/3

OTHER PUBLICATIONS

Savoy, J., Searching Information in Hypertext Systmes using Multiple Sources of Evidence, International Journal of Man–Machine Studies, 1993, pp. 1017–1030.

Stieger, H., Making Use of Hypertext Links when Retrieving Information, ACM, pp. 102–111, 1992.

R.W. Schwanke et al., Cross References are Features, Sec. 10.1, Book/Machine Learning: From Theory to Applications, Cooperative Research at Siemens and MIT, Appeared in Proceedings of the 2nd International Workshop on Software Configuration Mng., Princeton, NJ, Oct. 1989, ACM SIGSoft, IEEE CS, and GI pp. 107–123.

H.C. Arents et al., "Concept–Based Retrieval of Hypermedia Information: From Term Indexing to Semantic Hyperindexing," Information processing & Management vol. 29, No. 3, pp. 373–386, 1993.

R. Rada et al., "Retrieval Hierarchies in Hypertext, " Information Processing & Mng., vol. 29, No. 3, (Printed in Great Britain) pp. 359–371, 1993.

W.M. Shaw, Jr., "Subject and Citation Indexing. Part 1: The Clustering Structure of Composite Representations in the Cystic Fibrosis Document Collection, " JASIS–Journal of the American Society for Information Science, vol. 42, No. 9, Oct. 1991, pp. 669–675.

W.M. Shaw, Jr., "Subject Indexing & Citation Indexing–Part II: A Evaluation and Comparison Information Processing & Management", vol. 26, No. 6, (printed in Great Britain) pp. 705–718, 1990.

T.R. Kochtanek, "Brief Communication, Document Clustering, Using Macro Retrieval Techniques, " Journal of the American Society for Information Science, vol. 34, No. 5, pp. 356–359, Sep. 1993.

F. Narin et al., Chapter 2., "Bibliometrics, " Pub. Annual Review of Information Science and Technology, pp. 35–58, 1977.

(List continued on next page.)

Primary Examiner—Thomas G. Black
Assistant Examiner—John Loomis
Attorney, Agent, or Firm—Khanh Q. Tran

[57] **ABSTRACT**

A system and method are provided for searching for desired items from a network of information resources. In particular, the system and method have advantageous applicability to searching for World Wide Web pages having desired content. An initial set of pages are selected, preferably by running a conventional keyword-based query, and then further selecting pages pointing to, or pointed to from, the pages found by the keyword-based query. Alternatively, the invention may be applied to a single page, where the initial set includes pages pointed to by the single page and pages which point to the single page. Then, iteratively, authoritativeness values are computed for the pages of the initial set, based on the number of links to and from the pages. One or more communities, or "neighborhoods", of related pages are defined based on the authoritativeness values thus produced. Such communities of pages are likely to be of particular interest and value to the user who is interested in the keyword-based query or the single page.

57 Claims, 5 Drawing Sheets

Choose initial parameters:
m = number of initial pages
t = number of iterations
k = output size

Patent invented by Jon Kleinberg
Assigned to IBM; filed March 1997

Copyright Ellis Horowitz, 2011-2022

Cloud Computing and Assignment #3

Copyright Ellis Horowitz 2011-2022

Computing is Rapidly Changing

- **There are many trends putting pressure on conventional computing centers, e.g.**
  - Explosive growth in applications: biomedical informatics, space exploration, business analytics, web 2.0 social networking
  - Extreme scale content *generation*: e-science and e-business data deluge
  - Extraordinary rate of digital content *consumption*: digital gluttony: Apple iPhone, iPad, Amazon Kindle
  - Exponential growth in compute capabilities: multi-core, storage, bandwidth, virtual machines (virtualization)
  - Very short cycle of obsolescence in technologies: Windows Vista→ Windows 10 -> Windows 11; Java versions; JavaScript, C→C#; Python, Swift, Go, R
  - Newer architectures: web services, persistence models, distributed file systems/repositories (Google, Hadoop), multi-core, wireless and mobile
- **It is far more difficult for a company to manage this complex situation with a traditional IT infrastructure:**

Copyright Ellis Horowitz, 2011-2022

Enter the Cloud

- **Definition (Simple):** *Cloud computing* refers to the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer
- **Definition (Complicated):** *Cloud computing* is an information technology paradigm that enables ubiquitous access to shared pools of configurable system resources and higher-level services that can be rapidly provisioned with minimal management effort, often over the Internet.
- **Definition:** *Cloud computing* is Internet-based computing, whereby shared resources, software and information are provided to computers and other devices on-demand, like the electricity grid.
- **Other names for cloud computing:**
  - **on-demand computing, utility computing, ubiquitous computing, autonomic computing, platform computing, edge computing, elastic computing, grid computing, …**

Copyright Ellis Horowitz, 2011-2022

Cloud Computing Terminology

- **Infrastructure as a Service (IaaS)**
  - Virtual machine instances of various shapes and sizes
  - Storage capabilities
  - Networking support including configurable IP addresses
- **Platform as a Service (PaaS)**
  - Data management: text search, image analytics, deep learning
  - APIs for building applications
  - Business analytics
- **Software as a Service (SaaS)**
  - Enterprise resource planning software
  - Supply chain management software
- **Data as a Service (DaaS)**
  - Aggregate and analyze consumer data

Copyright Ellis Horowitz, 2011-2022

Cloud Service Models

Software as a Service (SaaS)	Platform as a Service (PaaS)	Infrastructure as a Service (IaaS)

SalesForce CRM

LotusLive

Software as a Service (SaaS) Providers Applications

Google App

Windows Azure
The Future Made Familiar

Platform as a Service (PaaS)

Deploy customer created Applications

amazon web services

rackspace HOSTING

Infrastructure as a Service (IaaS)

Rent Processing, storage, N/W capacity & computing resources

Copyright Ellis Horowitz 2011-2022

Cloud Computing Relies Upon the Internet, But . . .

- **Cloud Computing takes place over the Internet,**
- **These platforms hide the complexity and details of the underlying infrastructure from users and applications by providing a "simple" graphical interface or API**
- **However, the balkanization of the internet is also occurring**
  - ***Balkanization*** is a term for the process of fragmentation or division of a region or state into smaller regions or states that are often hostile or uncooperative with one another
  - **Balkanization is an effort to control what information is seen by the population, avoiding points of view that contradict those in power**
- **Splinternet:** a term used to describe efforts to control what people see on the internet (see https://en.wikipedia.org/wiki/Splinternet)
  - China's great firewall – Google withdrawing their search engine due to censorship requirements
  - Russia's Sovereign Internet Law allows for separation from the Internet
  - Iran, Syria, Tunisia, Saudi Arabia and others filter a wide range of internet content
  - South Korea filters news from North Korea

Copyright Ellis Horowitz, 2011-2022

Cloud Computing Platform is Aways on . . .

- **The platform provides on-demand services, that are always on, anywhere, anytime and any place**
  - Well almost, e.g. Amazon today (03/02/2017) blamed human error for the big AWS outage that took down a bunch of large internet sites for several hours on Tuesday afternoon
  - http://www.recode.net/2017/3/2/14792636/amazon-aws-internet-outage-cause-human-error-incorrect-command
  - https://techcrunch.com/2019/06/02/google-cloud-is-down-affecting-numerous-applications-and-services/
- **Pay for use and as needed, *elastic***
  - scale up and down in capacity and functionalities
- **The hardware and software services are available to everyone**
  - general public, enterprises, government

Copyright Ellis Horowitz, 2011-2022

Purpose and Benefits

- **By using the Cloud infrastructure on "pay as used and on demand", companies save in capital and operational investment!**
- **Clients can:**
  - Put their data on the platform instead of on their own desktop PCs and/or on their own servers.
  - Put their applications on the platform and use the servers within the cloud to do processing and data manipulations etc.
- **Enables companies and applications, which are system infrastructure dependent, to be infrastructure-less**
- **Cost-Benefit Analysis**
  - Will there be reduced IT costs
  - Will your applications be able to scale
  - Will you have access to automatic updates
  - Will your data be secure
  - There are other considerations

Copyright Ellis Horowitz, 2011-2022

Virtualization Makes Cloud Computing Possible

- **Virtualization**: the creation of a virtual -- rather than actual -- version of something, such as an operating system, a server, a storage device or network resources
- **Advantages of virtual machines:**
  1. Run operating systems where the physical hardware is unavailable,
  2. Easier to create new machines, backup machines, etc.,
  3. Software testing using "clean" installs of operating systems and software,
  4. Emulate more machines than are physically available,
  5. Timeshare lightly loaded systems on one host,
  6. Debug problems (suspend and resume the problem machine)
  7. Easy migration of virtual machines (shutdown needed or not)
  8. Run legacy systems!

Copyright Ellis Horowitz, 2011-2022

Hypervisor
(variant of supervisor)

- A **hypervisor** or **virtual machine monitor** (**VMM**) is computer software, firmware or hardware that creates and runs virtual machines
- A computer on which a hypervisor runs one or more virtual machines is called a *host machine*, and each virtual machine is called a *guest machine*.
- The hypervisor manages the execution of the guest operating systems.
- Multiple instances of a variety of operating systems may share the virtualized hardware resources: for example, Linux, Windows and MacOS, can all run on a single physical machine.
- Some hypervisor vendors
  - **VMware ESX Server,** ESX301
  - **Microsoft Windows Hyper-V**
  - **Oracle VM Virtual Box**
  - **Xen Project**
    - developed by the Univ. of Cambridge and is now being developed by the Linux foundation with support from INTEL XEN3030

Copyright E

Virtualization Makes Cloud Computing Possible

- **Virtual workspaces:**
  - An abstraction of an execution environment that can be made dynamically available to authorized clients by using well-defined protocols,
  - Resource quota (e.g. CPU, memory share),
  - Software configuration (e.g. O/S, provided services).
- **Virtual Machines (VMs):**
  - Abstraction of a physical host machine,
  - ***Hypervisor intercepts and emulates instructions from VMs, and allows management of VMs,***
  - VMWare, Xen, etc.
- **Provide infrastructure API:**
  - Plug-ins to hardware/support structures

Copyright Ellis Horowitz, 2011-2022

Virtualization Approaches

1. The **full virtualization** approach allows datacenters to run an unmodified guest operating system
   - **VMware** uses a combination of direct execution and binary translation techniques to achieve full virtualization of an x86 system
2. The **para-virtualization** approach modifies the guest operating system to eliminate the need for binary translation. Therefore it offers potential performance advantages for certain workloads but requires using specially modified operating system kernels
   - The **Xen open source project** was designed initially to support para-virtualized operating systems. While it is possible to modify open source operating systems, such as Linux and OpenBSD, it is not possible to modify "closed" source operating systems such as Microsoft Windows .

- Microsoft Windows is the most widely deployed operating system in enterprise datacenters.
  - For such unmodified guest operating systems, a virtualization hypervisor must either adopt the full virtualization approach or rely on hardware virtualization in the processor architecture.

Copyright Ellis Horowitz, 2011-2022

Hypervisor Features

Features

	Xen	KVM	VirtualBox	VMWare
Paravirtualization	Yes	No	No	No
Full Virtualization	Yes	Yes	Yes	Yes
Host CPU	X86, X86_64, IA64	X86, X86_64, IA64, PPC	X86, X86_64	X86, X86_64
Guest CPU	X86, X86_64, IA64	X86, X86_64, IA64, PPC	X86, X86_64	X86, X86_64
Host OS	Linux, Unix	Linux	Windows, Linux, Unix	Proprietary Unix
Guest OS	Linux, Windows, Unix	Linux, Windows, Unix	Linux, Windows, Unix	Linux, Windows, Unix
VT-x / AMD-v	Opt	Req	Opt	Opt
Supported Cores	128	16*	32	8
Supported Memory	4TB	4TB	16GB	64GB
3D Acceleration	Xen-GL	VMGL	Open-GL	Open-GL, DirectX
Licensing	GPL	GPL	GPL/Proprietary	Proprietary

Virtualization Tools

- **Most "Cloud" deployments rely on virtualization.**
  - Amazon EC2, GoGrid, Azure, Rackspace Cloud …
  - Nimbus, Eucalyptus, OpenNebula, OpenStack …
- **There are a number of Virtualization tools or Hypervisors available today.**
  - Xen, KVM, VMWare, Virtualbox, Hyper-V …

Leading Cloud Vendors

Cloud vendor	Annual revenue run rate
Microsoft commercial cloud	$21.2 billion
Amazon Web Services	$20.4 billion
IBM	$10.3 billion
Oracle	$6.08 billion
Google Cloud Platform/G Suite	$4 billion
Alibaba	$2.2 billion

Source: Company filings, earnings reports

Role of AI and Machine Learning

(All cloud vendors are emphasizing AI/ML)

- AWS offers multiple machine learning software e.g SageMaker, TensorFlow, PyTorch
- Google Cloud has BigQuery, AutoML, TensorFlow
- Microsoft emphasizes Azure Internet of Things capability, MLOps
- IBM offers its Watson platform
- Oracle offers automated cloud services

Copyright Ellis Horowitz 2011-2022

Comparing the Leading Cloud Vendors

- Amazon EC2 is at one end of the spectrum.
  - An EC2 instance looks much like physical hardware, and users can control nearly the entire software stack, from the kernel upwards
- Google AppEngine is at the other extreme of the spectrum
  - AppEngine is targeted exclusively at traditional web applications, enforcing an application structure of clean separation between a stateless computation tier and a stateful storage tier.
- Microsoft Azure
  - applications are written using the .NET libraries, and compiled to the Common Language Runtime, a language-independent managed environment. Thus, Azure is intermediate between application frameworks like AppEngine and hardware virtual machines like EC2.

Copyright Ellis Horowitz, 2011-2022

Computation Model

Examples of Cloud Computing vendors and how each provides virtualized resources (computation, storage, networking) and ensures scalability and high availability of the resources.

Computation Model (VM)

Amazon EC2

- x86 Instruction Set Architecture (ISA) via Xen VM.
- Computation elasticity allows scalability, but developer must build the machinery, or third party such as RightScale must provide it.

Microsoft's Azure

- Microsoft Common Language Runtime (CLR) VM; common intermediate form executed in managed environment.
- Machines are provisioned based on declarative descriptions (e.g. which "roles" can be replicated); automatic load balancing.

Google AppEngine

- Predefined application structure and framework; programmer-provided "handlers", written in Python, all persistent state stored in MegaStore (outside Python code).
- Automatic scaling up and down of computation and storage; network and server failover; all consistent with 3-tier Web app structure.

Copyright Ellis Horowitz, 2011-2022

Storage Model

Storage Model

Amazon EC2

- Range of models from block store (EBS) to augmented key/blob store (SimpleDB). Computation elasticity allows scalability, but developer must build the machinery, or third party such as RightScale must provide it.
- Automatic scaling varies from no scaling or sharing (EBS) to fully automatic (SimpleDB, S3), depending on which model used.
- Consistency guarantees vary widely depending on which model used.

Microsoft's Azure

- SQL Data Services (restricted view of SQL Server) .
- Azure storage service

Google AppEngine

- MegaStore/BigTable

Copyright Ellis Horowitz, 2011-2022

Now Let's Shift to HW#3 Redeeming Google Cloud Credits

Students will sign up for Google's free trial, $300 credit

Select Account Type Individual

use your @gmail.com address and finish by clicking "Start my free trial"

console.cloud.google.com

Google Cloud Platform Education Grants Credit

If you do not have a credit card, Google provides you with a coupon code via the Google Cloud Platform Education Grants program. If you do have a credit card, you can sign up for the Google Cloud Platform "Free Trial".

On Piazza and by e-mail, you will receive a communication like the one displayed in the image.

Click on the Student Coupon Retrieval Link. New window will open shown in the next slide.

Your Google Cloud Platform Education Grant for Students - University of Southern California — Inbox

---

Dear Students,

Here is the URL you will need to access in order to request a Google Cloud Platform coupon. You will be asked to provide your school email address and name. An email will be sent to you to confirm these details before a coupon is sent to you.

Student Coupon Retrieval Link

- You will be asked for a name and email address, which needs to match the domain. A confirmation email will be sent to you with a coupon code.
- You can request a coupon from the URL and redeem it until: 1/1/2019
- Coupon valid through: 9/1/2019
- You can only request ONE code per unique email address.

Please contact me if you have any questions or issues.

Google Cloud Platform Education Grants Credit

Enter your First Name, Last Name and your USC e-mail address. @usc.edu will be pre-filled. Click on Submit. If you entered a valid USC e-mail address, an email will be sent to that USC email address to verify that you own such address.

Once your USC email address is "verified", you will receive a second email with a Google Cloud Platform Coupon Code in it.

More details in Homework #3, Instructions for Setting Up Google Cloud Account documents (Refer 1.1)

Google Cloud Platform Education Grants Credit

- Before clicking on the link labeled in the email, you should open your default browser, and login to a Gmail account. Every USC student has been provided with a Gmail account. Once logged into Gmail, you can click on link in the mail , or you can go to this page: https://console.cloud.google.com/education to redeem your coupon. The web form below will be displayed.
- You need to paste your coupon into the field labeled Coupon code. Click on Accept and continue. You will now be taken to the Google Cloud Platform' s Billing section, and the amount of your credit will be displayed

Google Cloud Platform

Education grants

Please enter the coupon code provided to you via the Google Cloud Platform Education Grants program to receive credit for Google Cloud Platform. Get what you need to build and run your apps, websites and services.

Coupon code

[redacted]

Credit amount	Expiration date	Course
$100.00	Aug 22, 2019	CS572 Information Retrieval and Web Search Engines

Country of residence

United States

Please email me updates regarding feature announcements, performance suggestions, feedback surveys and special offers.

Yes No

I have read and agree to the GCP Marketplace Terms of Service.

Yes No

Google Cloud Platform education grants credits terms and conditions

By clicking "Accept and continue" below, you, on behalf of yourself and the organization you represent ("You") agree to these terms and conditions:

The credit is valid for Google Cloud Platform products and is subject to Your acceptance of the applicable Google Cloud Platform License Agreement and any other applicable terms of service. The credit is non-transferable and may not be sold or bartered. Unused credit expires on the date indicated on the media conveying the promotion code. The credit may be issued in increments as You use the credit over the period of time during which the credit is valid. Offer void where prohibited by law.

You represent that you are accepting the promotional credit on behalf of your educational institution and the credit can only be used on behalf of the educational entity and not for your personal use. You represent, on behalf of such educational entity, that (i) You are authorized to accept this credit; (ii) the credit is consistent with all applicable laws and regulations, including relevant ethics rules and laws; and (iii) the provision of credits will not negatively impact Google's current or future ability to do business with such educational entity.

You agree that we may share the following information with your educational institution and course instructor: (1) personal information that you provide to us during the coupon redemption process and (2) information regarding your use of the coupon and Google Cloud Platform products.

Accept and continue Clear

Google Cloud Home Page https://console.cloud.google.com

There are two menus available from the console: main and context

Main

context

tutorials

Google Cloud Main Menu

- From the hamburger menu in the top left corner, you can access a menu that brings you to the 5 major components of Google Cloud Platform (GCP):
- Compute
- Storage
- Stackdriver (company to manage distributed apps running on the cloud)
- Tools
- Big Data

- For this exercise you will use DataProc within BIG DATA to set up a cluster of compute instances

Context Menu

The context menu changes based on the current major component
Here are three examples

Snapchat was built on top of App Engine

App Engine lets clients host their software at datacenters managed by Google

App Engine is a Platform as a Service (PaaS) while Compute Engine is an Infrastructure as a Service (IaaS)

For the App Engine you just write your code and it automatically executes

- Snap (Snapchat) recently signed a $2 billion, five year contract with Google for its cloud services, which makes Snap Google's largest customer of its cloud platform
- Apple confirms it is using Google cloud for iCloud services

Create a Cluster

Using the **Google Cloud** Platform Console you can **create** a **cluster** by going to the **Cloud** Platform Console.

Select your project, and then click Continue to open the **Clusters** page.

Contains 1 master node and 3 worker nodes

Copyright Ellis Horowitz 2011-2022

Successful Creation of a Cluster

Cloud Dataproc

Clusters

Jobs

Clusters CREATE CLUSTER REFRESH DELETE

Name	Zone	Total worker nodes	Cloud Storage staging bucket	Created	Status
cluster-1	asia-east1-b	3	dataproc-723aedf5-c6dd-4fc7-b841-a1d731e6abaf-asia	Feb 19, 2017, 9:46:01 AM	Running

This URL will let you ssh to your cluster

Copyright Ellis Horowitz, 2011-2022

ssh into the Cluster

Dataproc

- Clusters
- Jobs
- Workflows
- Autoscaling policies
- Component exchange
- Notebooks

cluster-572-grader SUBMIT JOB REFRESH DELETE VIEW LOGS

For PD-Standard without local SSDs, we strongly recommend provisioning 1TB or larger to ensure consistently high I/O performance. See https://cloud.google.com/compute/docs/disks/performance for information on disk I/O performance.

Name	cluster-572-grader
Cluster UUID	706f39f6-ae82-4e2a-9d9d-4a872902454b
Type	Dataproc Cluster
Status	Running

MONITORING JOBS VM INSTANCES CONFIGURATION WEB INTERFACES

Filter instances

Name	Role	
cluster-572-grader-m	Master	SSH
cluster-572-grader-w-0	Worker	
cluster-572-grader-w-1	Worker	

Equivalent REST

Copyright Ellis Horowitz, 2011-2022

Environment Variables Set

Create a home directory

Check env variables
JAVA_HOME
HADOOP_CLASSPATH

ehorowitz1@cluster-1-m: ~

Secure https://ssh.cloud.google.com/projects/horowitz-project/zones/asia-east1-b/instances/cluster-1-m?authuser=0&hl=en_US&...

```
ehorowitz1@cluster-1-m:~$ env
TERM=xterm-256color
SHELL=/bin/bash
SSH_CLIENT=173.194.90.33 63226 22
SSH_TTY=/dev/pts/0
USER=ehorowitz1
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:su=37;41:sg=3
0;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha
=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*
.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.t
bz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;3
1:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp
=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35
:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.we
bm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;3
5:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=0
1;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.f
lac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;
36:*.axa=00;36:*.oga=00;36:*.spx=00;36:*.xspf=00;36:
DATAPROC_MASTER_HA_COMPONENTS=hadoop-hdfs-journalnode hadoop-hdfs-zkfc zookeeper-server
SSH_AUTH_SOCK=/tmp/ssh-zGxHCr3rJ9/agent.3623
DATAPROC_MASTER_COMPONENTS=hadoop-hdfs-namenode hadoop-yarn-resourcemanager mysql-server
MAIL=/var/mail/ehorowitz1
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
PWD=/home/ehorowitz1
JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
HADOOP_CLASSPATH=/lib/tools.jar
LANG=en_US.UTF-8
DATAPROC_COMMON_COMPONENTS=openjdk-8-jdk libjansi-java python-numpy libmysql-java hadoop-client hive pig spark-core
 spark-python spark-r autofs nfs-common libhdfs0 libsnappy1 libatlas3-base libopenblas-base libapr1 vim git bash-co
mpletion spark-yarn-shuffle spark-datanucleus spark-extras hadoop-lzo
DATAPROC_MASTER_STANDALONE_COMPONENTS=hadoop-hdfs-secondarynamenode
ALPN_JAR=/usr/local/share/google/alpn/alpn-boot-8.1.7.v20160121.jar
DATAPROC_WORKER_COMPONENTS=hadoop-hdfs-datanode hadoop-yarn-nodemanager
SHLVL=1
HOME=/home/ehorowitz1
BDUTIL_DIR=/usr/local/share/google/dataproc/bdutil-dataproc-20170214-094528-RC1
LOGNAME=ehorowitz1
SSH_CONNECTION=173.194.90.33 63226 10.140.0.4 22
DATAPROC_AGENT_JAR=/usr/local/share/google/dataproc/agent-20170214-094528-RC1.jar
DATAPROC_MASTER_EXCLUSIVE_COMPONENTS=hadoop-mapreduce-historyserver hive-metastore hive-server2 nfs-kernel-server s
park-history-server
_=/usr/bin/env
```

Disable Billing for Your Cluster

- Please **disable** the billing for the cluster when you are not using it.
- Leaving it running will cost extra credits.
- The cluster is billed based on how many hours it is running and not how much data it is processing

Copyright Ellis Horowitz 2011-2022

Upload the Data Set

- **We'll be using a collection of 74 files of web pages whose HTML has been removed**
  - The data comes from https://ebiquity.umbc.edu/resource/html/id/351
  - The data has been cleaned of metadata, license information, notes
- **Retrieve the dataset either from the cloud or from the CS572 website**
  - http://csci572.com/2022Fall/hw3/DATA.zip
  - https://drive.google.com/drive/folders/1Z4KyalIuddPGVkIm6dUjkpD_FiXyNIcq
- **Unzip the contents**
  - two folders inside named 'development' and 'full data'.
  - Each of the folders contains the actual data (web page content) and a mapper file to map the docID to the file name.

Copyright Ellis Horowitz 2011-2022

Uploading data to GCP

- **You can upload data to GCP using the GUI, mentioned in the document Hadoop Exercise to Create an Inverted Index:**

  http://csci572.com/2022Fall/hw3/HadoopExercise.pdf

  -OR-
- **Install Google Cloud SDK Shell using this link:**

  https://cloud.google.com/storage/docs/gsutil_install
- **Select your operating system from the options provided**
- **Follow the instructions mentioned**
- **Upload data using:**

```
gsutil –m cp –r "\Users\Documents\fullData" "gs://bucket/foldername"
```

- ***Instead of** `\Users\Documents\fullData` **provide the path on your computer to data.**
- ***Instead of** `gs://bucket/foldername` **provide the directory on your GCP storage**
  - **Eg:** `gs://dataproc-e2659b1-9ce7-4d18-93b1-83c013225-us-west1/Data`
- **An example command is:**

```
gsutil -m cp -r "C:\Users\IR\Docs\Data\devdata" "gs://dataproc-e2659b1-9ce7-4d18-93b1-83c013225-us-west1/Data"
```

Development Data

DATA	--	Folder	Today at 9:35 AM
devdata	--	Folder	Today at 9:35 AM
5722018101.txt	43.6 MB	Plain Text	Today at 9:35 AM
5722018235.txt	41.8 MB	Plain Text	Today at 9:35 AM
5722018301.txt	36.4 MB	Plain Text	Today at 9:35 AM
5722018496.txt	12.8 MB	Plain Text	Today at 9:35 AM
5722018508.txt	52.4 MB	Plain Text	Today at 9:35 AM
fulldata	--	Folder	Today at 9:35 AM
5722018435.txt	47.3 MB	Plain Text	Today at 9:35 AM
5722018436.txt	47.6 MB	Plain Text	Today at 9:35 AM
5722018437.txt	49.1 MB	Plain Text	Today at 9:35 AM
5722018438.txt	41.6 MB	Plain Text	Today at 9:35 AM
5722018439.txt	44.5 MB	Plain Text	Today at 9:35 AM
5722018440.txt	46.6 MB	Plain Text	Today at 9:35 AM

5722018101.txt

```
5722018101      "The DeLorme PN-20 represents a new breed of GPS devices. ...a fantastic
device, and it leads the way in a new breed of GPS devices which can display aerial
photography and satellite imagery. For people who have dreamed about having a Google Earth
type product in a handheld device... this is it."November 25, 2005 marked Something
Fishy's ten year anniversary on the web! We are one of the largest, oldest and most
comprehensive web sites available on the Anorexia, Bulimia, Compulsive Overeating and
Binge Eating Disorders, providing information and support to sufferers and their loved
ones.Do you know your family members? You might not know them as well as you think. We
gathered a listing of comments from members about what they are really feeling and what
they wished their friends and family really knew about them. If You Really Knew Me...Our
comprehensive eating disorders treatment finder at Something Fishy contains listings from
over 1,800 therapists, dieticians, treatment centers and other professionals worldwide
working to help those with Anorexia, Bulimia, Compulsive Overeating and Binge Eating
Disorder recover. Fully searchable by category (type of treatment), country, state, area
code, name, services, description or zipcode.Our Mission: We are dedicated to raising
awareness about eating disorders... emphasizing always that eating disorders are NOT about
food and weight; They are just the symptoms of something deeper going on, inside.
Something Fishy is determined to remind each and every sufferer of anorexia, bulimia,
compulsive overeating and binge eating disorder that they are not alone, and that complete
recovery is possible. If you are the loved-one of someone that suffers with an eating
disorder, use this website to educate yourself. The more you know, the more you are
equipped to provide the support your loved-one needs. If you have an eating disorder, you
can find help. You can recover. And you deserve to do both.Though our site should be
friendly to most browsers it is best viewed on Internet Explorer 4 (or higher) or Netscape
```

REMEMBER TO disable the billing for the cluster when you are not using it

Click here and select Billing

Google Cloud Platform
Search products and resources

Billing
Account man...
Information Retrieval and Web Search Engines
CHANGE ORGANIZATION
RENAME BILLING ACCOUNT
HIDE INFO PANEL

- Overview
- Reports
- Cost table
- Cost breakdown
- Commitments
- Budgets & alerts
- Billing export
- Pricing
- Account management

Billing account ID
01B33E-145B36-A9A5A1

Projects linked to this billing account

Project name	Project ID	Actions
csci572-hw3	csci572-hw3-270906	⋮

Disable billing
Change billing

Information R...

Edit or delete permissions below or "Add Member" to grant new
ADD MEMBER
Show inherited permissions

Filter tree

Role / Member
Billing Account Administrator (1)

Inverted Index Implementation

- **You need to write some code, in Java, that processes the data file of web pages and produces an inverted index of the words that occur**
- **Google Cloud requires the code to be packaged as a jar file, e.g.**
- **If your Java program is called** `InvertedIndexJob.java`
  - **first compile the code and then**
  - **run the jar program**
- `hadoop com.sun.tools.javac.Main InvertedIndexJob.java`
- `jar cf invertedindex.jar InvertedIndex*.class`
- **Place this jar file in the default cloud bucket of your cluster in a folder called JAR on your bucket and upload it to that folder**

Copyright Ellis Horowitz, 2011-2022

Mapper Class

The Google cluster requires that you write two routines to implement the Map/Reduce functionality

A lecture on Map/Reduce is coming later

Class is WordCountMapper; *Routine* is map(key,value,context)

Program reads a line of text, and for each token (word) that it finds on the line it sends the pair (token, 1) to the collector/ reducer

```
/*
This is the Mapper class. It extends the Hadoop's Mapper class.
This maps input key/value pairs to a set of intermediate(output) key/value pairs.
Here our input key is a LongWritable and input value is a Text.
And the output key is a Text and value is an IntWritable.
*/
class WordCountMapper extends Mapper<LongWritable, Text, Text, IntWritable>
{
    /*
    Hadoop supported data types. This is a Hadoop specific datatype that is used to handle
    numbers and Strings in a hadoop environment. IntWritable and Text are used instead of
    Java's Integer and String datatypes.
    Here 'one' is the number of occurances of the 'word' and is set to the value 1 during the
    Map process.
    */
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(LongWritable key, Text value, Context context)
        throws IOException, InterruptedException
    {
        //Reading input one line at a time and tokenizing.
        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);

        //Iterating through all the words available in that line and forming the key value pair.
        while (tokenizer.hasMoreTokens())
        {
            word.set(tokenizer.nextToken());
            /*
            Sending to output collector(Context) which in-turn passes the output to Reducer.
            The output is as follows:
                'word1' 1
                'word1' 1
                'word2' 1
            */
            context.write(word, one);
        }
    }
}
```

Reducer Class

Class is `WordCountReducer`; *Program* is `reduce(key, values,context)`

For each key (word), the number of occurrences are summed together and written out

```
/*
This is the Reducer class. It extends the Hadoop's Reducer class.
This maps the intermediate key/value pairs we get from the mapper to a set
of output key/value pairs, where the key is the word and the value is the word's count.
Here our input key is a Text and input value is a IntWritable.
And the output key is a Text and value is an IntWritable.
*/
class WordCountReducer extends Reducer<Text, IntWritable, Text, IntWritable>
{
    /*
    Reduce method collects the output of the Mapper and adds the 1's to get the word's count.
    */
    public void reduce(Text key, Iterable<IntWritable> values, Context context)
            throws IOException, InterruptedException
    {
        int sum = 0;
        /*
        Iterates through all the values available with a key and add them together and give the
        final result as the key and sum of its values
        */
        for (IntWritable value : values)
        {
            sum += value.get();
        }
        context.write(key, new IntWritable(sum));
    }
}
```

Copyright Ellis Horowitz 2011-2022

Main Class

Class `WordCount`;
Program: `main`
Create the Hadoop job

```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.*;
public class WordCount
{
    public static void main(String[] args)
        throws IOException, ClassNotFoundException, InterruptedException {
        if (args.length != 2) {
            System.err.println("Usage: Word Count <input path> <output path>");
            System.exit(-1);
        }
        //Creating a Hadoop job and assigning a job name for identification.
        Job job = new Job();
        job.setJarByClass(WordCount.class);
        job.setJobName("Word Count");
        //The HDFS input and output directories to be fetched from the Dataproc job submission console.
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        //Providing the mapper and reducer class names.
        job.setMapperClass(WordCountMapper.class);
        job.setReducerClass(WordCountReducer.class);
        //Setting the job object with the data types of output key(Text) and value(IntWritable).
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        job.waitForCompletion(true);
    }
}
```

Copyright Ellis Horowitz 2011-2022

Built-in Tutorials

- Click the tricolon at the top right of the console and select "*Try an interactive tutorial*" to be brought to this list of tutorials
- An advantage of these built-in tutorials is they'll step you through each piece and do necessary project management. It's fine to use the default first project for all of these.

Sample Document and Output

5722018411 A look at the most publicized aspects of the strike-- economics, stress, and management-- shows how these issues obscured and distorted the controllers' main concern of workplace control and helps explain why problems persist in the ATC workforce. It also demonstrates how management and labor's focus on economic issues since World War II has bankrupted labor's discourse and limited its ability to address concerns outside of a narrow range of concerns.These perceptions were in part responsible for the overwhelming public approval of Reagan's handling of the strike; 65% in a public opinion poll; mail, according to one representative, ran 1000 to 1 in favor of the administration.Most strikers denied that money was a critical component in their decision to strike. Yet Poli insisted that his demands, headed by a pay raise, reflected the desires of his constituency. Arthur Shostak, who conducted five surveys of PATCO members in 1979 and 1980 backs up Poli's assertion that salary was important to the strikers. It is tempting to concede then that workers did see the

Sample of Mapper Output

```
aspect 5722018411
distorted 5722018411
economics 5722018411
economics 5722018411
management 5722018411
publicized 5722018411
```

Sample of Reducer Output

```
answer  5722018453:2 5722018483:1
antecedence 5722018502:1 5722018435:1
asterisks.  5722018417:1 5722018504:2 5722018447:1
beautiful   5722018439:7 5722018417:2 5722018416:3 5722018438:5 5722018437:1 5722018415:1 5722018414:2 5722018435:3
bind    5722018419:6 5722018417:39 5722018416:1
chunking    5722018507:1 5722018502:1
```

aspect occurred 1 time in the document with docID 5722018411

economics occurred 2 times in the document with docID is 5722018411

answer appears twice

Copyright Ellis Horowitz 2011-2022

Credits Spent

- To check how much you've been charged for your cluster,
  - navigate to the Billing section and
  - click on the project name in the Overview section
  - check this section at least once every 24 hours.

Billing | Overview | CS572 Information Retrieval | RENAME BILLING ACCOUNT

- Overview
- Budgets & alerts
- Billing export

Billing account ID: 00AB19-9B02F2-9EA431

Credits

Promotion ID	Expires	Promotion value	Amount remaining
CS572 Information Retrieval	Jan 9, 2018	$150.00	$92.16

Projects linked to this billing account

Project name	Project ID
My First Project	euphoric-drive-158517

Billing | Charges this month

Go to billing

Product	Resource	Usage	Amount
Google Compute	Standard Intel N1 4 VCPU running in JP	11,280 Minutes	$49.86
Google Compute	Google Cloud Dataproc running on VM Image CORE	45,120 Minutes	$7.52
Google Compute	Storage Pd Capacity Jp	8.95 GB-month	$0.47
Google Compute	Storage Pd Capacity Jp	Credit applied	$-0.47
Google Compute	Google Cloud Dataproc running on VM Image CORE	Credit applied	$-7.52
Google Compute	Standard Intel N1 4 VCPU running in JP	Credit applied	$-49.86

*Estimated charges before taxes, updated daily — Total: $0.00

Copyright Ellis

Google Building Blocks: MapReduce Google File System BigTable

These slides borrow material from Mining Massive Datasets by Rajaraman and Ullman and class notes by Rajaraman (Stanford) and Weld (U. Washington) , D. Weld (Google) and S. Ghemawat (Google), Vera Goebel (Univ. of Oslo)

Copyright Ellis Horowitz 2011-2022

Google Specialized Software Systems

- **Google has built several major software systems for their internal processing**
  1. **MapReduce - an easy way to write and run large-scale jobs on clusters of machines**
     - generate production index data more quickly
     - perform ad-hoc experiments rapidly
     - Dean & Ghemawat. *MapReduce: Simplified Data Processing on Large Clusters*, OSDI, 2004
  2. **GFS (Google File System) a large-scale distributed file system**
     - Ghemawat, Gobioff, & Leung. *Google File System*, SOSP 2003
  3. **BigTable - a semi-structured storage system**
     - online, efficient access to per-document information at any time
     - multiple processes can update per-doc info asynchronously
     - critical for updating documents in minutes instead of hours
     - Chang, Dean, Ghemawat, Hsieh, Wallach, Burrows, Chandra, Fikes, & Gruber. *Bigtable: A Distributed Storage System for Structured Data*, OSDI 2006

Copyright Ellis Horowitz, 2011-2022

Introduction to MapReduce

- MapReduce is a methodology for exploiting parallelism in computing clouds (racks of interconnected processors)
- It has become a common way to analyze very large amounts of data
- MapReduce was initially developed at Google
- In 2004 Google was using MapReduce to process 100TB/day of data
- Today there are MapReduce Users Groups around the world, see https://www.meetup.com/topics/mapreduce/

Copyright Ellis Horowitz, 2011-2022

How is MapReduce Used by Search Engines?

- **At Google:**
  - Building Google's Search Index
  - Article clustering for Google News
  - Statistical machine translation
- **At Yahoo!:**
  - Building Yahoo!'s Search Index
  - Spam detection for Yahoo! Mail
- **At Facebook:**
  - Data mining
  - Ad optimization
  - Spam detection

Copyright Ellis Horowitz, 2011-2022

Motivation Beyond Search Engines

- **Modern Internet applications have created a need to manage immense amounts of data quickly.**
- **In many of these applications, the data is extremely regular, and there is ample opportunity to exploit parallelism.**
- **Some examples**

1. **Dish network collecting every click of the remote**
   - Dish network supplies TV reception via satellite; they collect data on their set top box and send it back to headquarters
2. **Tesla collecting every usage of the car**
   - Tesla's are connected to the cellular network; the car reports back all of its actions to Tesla

Copyright Ellis Horowitz, 2011-2022

Why Parallelization is Hard

- **Parallelization is "easy" if processing can be cleanly split into *n* units:**

- **Assigned to *n* workers**
- **And there is an easy way to combine the outputs**

Why Parallelization is Hard

we would like to have as many threads as there are work units, but this may not be the case

But there are complicated issues to deal with!

- **What if we have more work units than threads?**
- **How do we assign work units to worker threads?**
- **How do we aggregate/combine the results at the end?**
- **How do we know all the workers have finished?**
- **What if the work cannot be divided into completely separate tasks?**

- ***MapReduce solves all of these problems so the programmer does not have to deal with them***

Copyright Ellis Horowitz, 2011-2022

Programming with Multiple Threads Poses Challenges

Life Cycle of a Thread

Thread 1:	**Thread 2:**
void foo() {	void bar() {
x++;	y++;
y = x;	x++;
}	}

If the initial state is x = 6, y = 0, what are the final values of x and y after the threads finish running? Possible solutions include: (8,8) and (8,7)

Copyright Ellis Horowitz, 2011-2022

Multithreaded = Unpredictability

- **Many things that look like "one step" operations actually take several steps under the hood:**

Thread 1:

```
void foo() {
  eax = mem[x];
  inc eax;
  mem[x] = eax;
  ebx = mem[x];
  mem[y] = ebx;
}
```

Thread 2:

```
void bar() {
  eax = mem[y];
  inc eax;
  mem[y] = eax;
  eax = mem[x];
  inc eax;
  mem[x] = eax;
}
```

the same programs, but at the assembly code level

- **When we run a multithreaded program, we don't know what order threads run in, nor do we know when they will interrupt one another.**

Copyright Ellis Horowitz, 2011-2022

The "corrected" example

Thread 1:

```
void foo() {
  sem.lock();
  x++;
  y = x;
  sem.unlock();
}
```

Thread 2:

```
void bar() {
  sem.lock();
  y++;
  x++;
  sem.unlock();
}
```

The global variable `sem`, as defined by

```
Semaphore sem = new Semaphore();
```

guards access to x & y; semaphores are generally integer variables that are shared between threads; the variable protects the "critical section" from being simultaneously accessed

Copyright Ellis Horowitz, 2011-2022

Processing Across a Machine Cluster Introduces Unpredictability on Many Levels

- **Synchronization problems apply to more than just low level operations within a critical section of code**
  - **Other issues include:**
  - **Pulling work units from a queue**
  - **Assigning work units to an available thread**
  - **Work units reporting back to the master unit**
  - **Telling another thread that it can begin the "next phase" of processing**

… All require synchronization!

Copyright Ellis Horowitz, 2011-2022

How MapReduce Solves the Parallelization Problems

- **So MapReduce provides**
  - Automatic parallelization of code across multiple threads and across multiple processors
  - Fault tolerance in the event of failure of one or more nodes
  - I/O scheduling
  - Monitoring & Status updates

Copyright Ellis Horowitz, 2011-2022

Map/Reduce - Beginnings

- **Map/Reduce**
  - Is a programming model borrowed from the programming language Lisp
  - (and other functional languages, e.g. ML*)
- **Many problems can be phrased using the MapReduce paradigm**
- **Easy to distribute computation across nodes**
- **Nice retry/failure semantics**

*ML programming language is a general purpose functional programming language, see **https://en.wikipedia.org/wiki/ML_(programming_language)**

Copyright Ellis Horowitz, 2011-2022

Map and Reduce Functions in LISP (Scheme is a dialect of Lisp)

- (map *f list [list$_2$ list$_3$ …]*) General formulation
- Specific example
  - (map square ‘(1 2 3 4))
  - (1 4 9 16)

- (reduce *f id list*)
- Specific example
  - (reduce + 0 ‘(1 4 9 16))
  - (+ 16 (+ 9 (+ 4 (+ 1 0)))
  - 30

Copyright Ellis Horowitz, 2011-2022

What is MapReduce?

- **MapReduce is a programming model that generically works this way:**
  - **A *map function* extracts some intelligence from raw data**
  - **A *shuffle step* organizes the resulting output**
  - **A *reduce function* aggregates the data output from the shuffle step**
  - **Users specify the computation in terms of a *map* and a *reduce* function,**
  - **Underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, and**
  - **Underlying system also handles machine failures, efficient communications, and performance issues.**

-- Reference: Dean, J. and Ghemawat, S. 2008 "MapReduce: Simplified Data Processing on Large Clusters", *Communication of ACM* 51, 1 (Jan. 2008), 107-113.

Copyright Ellis Horowitz, 2011-2022

The Map/Reduce Paradigm

1. **A large number of records are broken into segments**
2. **Map: extracts something of interest from each segment**
3. **Group: sorts the intermediate results from each segment (sometimes called shuffle)**
4. **Reduce: aggregates intermediate results**
5. **Generate final output**

Key idea: to re-phrase problems in such a way that the input can be divided into parts and operated on in parallel and the results combined to produce a solution to the original problem

Copyright Ellis Horowitz, 2011-2022

The Map&Reduce Routines

- Using map-reduce one must write 2 functions called *Map* and *Reduce*
- The system manages the parallel execution and coordination of tasks; it is all done automatically
- A map-reduce computation proceeds as follows:
  1. Some number of map tasks each are given one or more chunks to process
  2. These map tasks turn the chunk into a sequence of key-value pairs; the way the pairs are produced depends upon the code for the Map function
  3. Key-value pairs from each Map task are collected by a master controller and sorted by key; keys are divided among all the Reduce tasks, so all key-value pairs with the same key wind up at the same Reduce task
  4. Reduce tasks work on one key at a time, and combine all the values associated with that key in some way; the manner of combination depends upon the Reduce code

Copyright Ellis Horowitz, 2011-2022

Schematic of a Map-Reduce Computation

Copyright Ellis Horowitz 2011-2022

A MapReduce Example – Counting Word Occurrences

- Counting the number of occurrences for each word in a collection of documents
- The input file is a repository of documents
- Each document is an element passed to a separate processor
- The Map function
  - Parses the document, extracts each word and uses each word as a key of type String (the words obtained by parsing), $w_1, w_2, \ldots$
  - For each word it assigns an integer, 1;
  - Each processor outputs key-value pairs where the key is a word and the value is always 1, namely $(w_1,1), (w_2,1),\ldots,(w_n,1)$
- If a word w appears n times in a single document, then there will be n key-value pairs $(w,1)$ in the output of the processor handling that document
- If a word w appears m times among all documents, then there will be m key-value pairs $(w,1)$ in the output

Copyright Ellis Horowitz, 2011-2022

Count Word Occurrences Pseudo-Code

- **The code below is similar to what a programmer would write to process multiple documents on a cluser of machines using map/reduce**

```
Map(String input_key, String input_value):
// input_key: document name
// input_value: document contents
for each word w in input_value:
   EmitIntermediate(w, "1");

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

Copyright Ellis Horowitz 2011-2022

Word Count Execution
An Alternate Graphical View

Distributed Execution Overview

see the next slide for an explanation

Copyright Ellis Horowitz, 2011-2022

Looking Under the Hood at the Map Task

- The user program forks a *Master* controller process and some number of *Worker* processes at different compute nodes;
- A *Worker* handles either Map tasks or Reduce tasks, but not both
- The *Master* must
  - Create some number of Map and Reduce tasks
  - These tasks are assigned to *Worker* processes by the *Master*
  - Typically there is one Map task for every chunk of the input
  - Keeps track of the status of each Map and Reduce task (states are: idle, executing on a Worker, completed)
- Each Map task is assigned one or more chunks of the input file(s) and executes on it the code
- The Map task creates a file for each Reduce task on the local disk of the Worker that executes the Map task
- The *Master* is told of the location and sizes of each of these files and the Reduce task for which each is destined

Copyright Ellis Horowitz, 2011-2022

Looking Under the Hood at the Reduce Task

- The *master controller* process knows how many Reduce tasks there will be, say *r*
- The user defines *r*
- The master controller picks a hash function that applies to keys and produces a bucket number from *0* to *r-1*
- Each key output by a Map task is hashed and its key-value pair is put in one of *r* local files
  - Each file will be processed by a Reduce task
- After all Map tasks have completed successfully, the master controller merges the file from each Map task that are destined for a particular Reduce task and feeds the merged file to that process
- For each key *k*, the input to the Reduce task that handles key *k* is a pair $(k, [v_1,\ldots,v_n])$ where $(k,v_1),(k,v_2),\ldots,(k,v_n)$ are all the key-value pairs with key *k* coming from all the Map tasks

Copyright Ellis Horowitz, 2011-2022

Explanation of the Reduce Task

- The *Reduce function* is written to take pairs consisting of a key and a list of associated values, and combines them in some way
- The *Reduce function* output is a sequence of key-value pairs consisting of each input key *k* paired with the combined value
- Outputs from all Reduce tasks are merged into a single file
- *Reduce function* adds up all the values and outputs a sequence of *(w,m)* pairs where *w* is a word that appears at least once in the documents and *m* is the total number of occurrences
- The *Reduce function* is generally associative and commutative implying values can be combined in any order yielding the same result

Copyright Ellis Horowitz, 2011-2022

Fault Tolerance in MapReduce

1. **If a task crashes:**
   - Retry on another node
     - OK for a map because it had no dependencies
     - OK for reduce because map outputs are on disk
   - If the same task repeatedly fails, fail the job or ignore that input block
2. **If a node crashes:**
   - Relaunch its current tasks on other nodes
   - Relaunch any maps the node previously ran
     - Necessary because their output files were lost along with the crashed node
3. **If a task is going slowly (straggler):**
   - Launch second copy of task on another node
   - Take the output of whichever copy finishes first, and kill the other one

Copyright Ellis Horowitz, 2011-2022

Coping with Node Failure

- Worst case: **the compute node where the Master is executing fails**
  - **Result**: the entire map-reduce job must be restarted
- Other failures are less severe and are handled by the Master
- **The compute node of a Map worker fails**
  - This is detected by the Master and all Map tasks that were assigned are re-done
  - The Master sets the status of each Map task to idle and re-schedules them when a worker becomes available
  - The Master informs each Reduce task of the location of its new input
- **The compute node of a Reduce worker fails**
  - The Master sets the status of its currently executing Reduce tasks to idle and they will be re-scheduled on another reduce worker later

Copyright Ellis Horowitz, 2011-2022

Typical Data Center Cluster

Characteristics of a Google Data Center

1. Google data centers (approx. two dozen): They come online and automatically, under the direction of the Google File System, start getting work from other data centers. These facilities, sometimes filled with 10,000 or more Google computers, find one another and configure themselves with minimal human intervention.
2. Standard desktop PCs: The hardware in a Google data center can be bought at a local computer store.
3. Each Google server comes in a standard case called a *pizza box* with one important change: the plugs and ports are at the front of the box to make access faster and easier.
4. Google racks are assembled for Google to hold servers on their front and back sides. This effectively allows a standard rack, normally holding 40 pizza box servers, to hold 80 servers.
5. A Google data center can go from a stack of parts to online operation in as little as 72 hours, unlike more typical data centers that can require a week or even a month to get additional resources online.
6. Each server, rack and data center works in a way that is similar to what is called "plug and play." Like a mouse plugged into the USB port on a laptop, Google's network of data centers knows when more resources have been connected. These resources, for the most part, go into operation without human intervention.

Copyright Ellis Horowitz, 2011-2022

Typical Cluster Architecture

- **Each rack of cpu's contains between 16-64 nodes**
- **Nodes within a single rack are connected by gigabyte Ethernet**
- **Each rack is connected to another rack by a switch with speeds of 2-10 Gbps**
- **Individual cpu's can fail; switches between racks can fail**

Typical Cluster Machine Configuration

Copyright Ellis Horowitz, 2011-2022

The Problems Google Tried to Solve with a New File System

- Google needed a large-scale and high-performance unified storage system that must:

1. **Be global**. Any client can access (read/write) any file. This allows for sharing of data among different applications.
2. **Support automatic sharding** of large files over multiple machines. This improves performance by allowing parallel processes on each file chunk and also deals with large files that cannot fit into a single disk.
3. **Support automatic recovery** from failures.
4. **Be optimized for sequential access** to huge files and for read and append operations which are the most common.

Copyright Ellis Horowitz, 2011-2022

Google File System General Goals

- **A scalable, distributed file system for large distributed data-intensive applications**
  - Provides fault tolerance
  - Runs on cheap, commodity hardware
  - Delivers high aggregate performance to large number of clients
- **GFS: not your typical file system**
  - Lacks typical per-directory data structure to list each file in the directory
  - Does not support aliases (i.e. hard or sym links)
  - Namespace: lookup table that maps full pathnames to metadata
  - Lookup table fits in memory (prefix compression)
    - Also known as incremental encoding is a type of delta encoding **compression** algorithm whereby common **prefixes** and their lengths are recorded so that they need not be duplicated
    - https://en.wikipedia.org/wiki/Incremental_encoding

Copyright Ellis Horowitz, 2011-2022

The Google File System Design Assumptions –

- **Files will be HUGE**
- **Multi-gigabyte files are common**
  - Not practical to have ~8 billion files
  - Each file contains many "application objects"
- **Multi-terabyte datasets**
  - I/O operations, block sizes must be considered
- **Most file modifications are appends (atomic)**
  - Random writes practically non-existent
  - Once written… sequential reads
  - Caching not terribly important
- **there will be a single master server monitoring multiple chunk servers**

Copyright Ellis Horowitz, 2011-2022

Google File System Top Level View

- **Google File System (GFS)**, is a proprietary distributed file system for efficient, reliable access to data using large clusters of commodity hardware
- Files are divided into fixed-size *chunks* of 64 megabytes, similar to clusters or sectors in regular file systems, which are only extremely rarely overwritten, or shrunk; files are usually appended to or read

GFS is designed for system-to-system interaction; Chunk servers replicate the data automatically
See https://en.wikipedia.org/wiki/Google_File_System

Copyright Ellis Horowitz, 2011-2022

Master Server and Chunk Servers

- **Master Server** holds all metadata:
  - Namespace (directory hierarchy)
  - Access control information (per-file)
  - Mapping from files to chunks
  - Current locations of chunks (chunkservers)
- Delegates consistency management
- Garbage collects orphaned chunks
- Migrates chunks between chunk servers

- **Chunk Server**
- Stores 64 MB file chunks on local disk using standard Linux filesystem, each with version number and checksum
- Read/write requests specify chunk handle and byte range
- Chunks replicated on configurable number of chunkservers (default: 3)
- No caching of file data (beyond standard Linux buffer cache)

Copyright Ellis Horowitz, 2011-2022

Google File System (GFS)

- Master manages metadata
- Data transfers happen directly between clients/chunkservers
- Files broken into chunks (typically 64 MB)
- Chunks triplicated across three machines for safety

Copyright Ellis Horowitz, 2011-2022

GFS: Major Aspects

- **Append vs. Rewrite**
  - GFS is optimized for appended files rather than rewrites. That's because clients within Google rarely need to overwrite files -- they add data onto the end of files instead. While it's still possible to overwrite data on a file in the GFS, the system doesn't handle those processes very efficiently
- **Which Replica Does GFS use?**
  - The GFS separates replicas into two categories: **primary replicas** and **secondary replicas**. A primary replica is the chunk that a chunkserver sends to a client. Secondary replicas serve as backups on other chunkservers.
  - The master server decides which chunks will act as primary or secondary. If the client makes changes to the data in the chunk, then the master server lets the chunkservers with secondary replicas know they have to copy the new chunk off the primary chunkserver to stay current.
- **What About Big Files?**
  - If a client creates a write request that affects multiple chunks of a particularly large file, the GFS breaks the overall write request up into an individual request for each chunk. The rest of the process is the same as a normal write request.
- **Heartbeats and Handshakes**
  - The GFS components give system updates through electronic messages called **heartbeats** and **handshakes**. These short messages allow the master server to stay current with each chunkserver's status.

Copyright Ellis Horowitz, 2011-2022

Google File System vs. BigTable

- **GFS provides raw data storage**
- **But Google needs a system for handling:**
  - Trillions of URLs
  - Geographic locations such as physical entities, roads, satellite image data, etc
  - Per user data for billions of people including preference settings, recent queries and searches
  - And it must be capable of
    - storing semi-structured data
    - Reliable, scalable, etc
- **Bigtable is a compressed, high performance, proprietary data storage system built on top of the Google File System**
- **It is used by a number of Google applications, such as web indexing, MapReduce, Google Maps, YouTube and Gmail**

Copyright Ellis Horowitz, 2011-2022

Big Table Data Model

- Not a Full Relational Data Model
- Provides a simple data model
  - Supports dynamic control over data layout
  - Allows clients to reason about the locality properties
- A Table in Bigtable is a:
  - Sparse
  - Distributed
  - Persistent
  - Multidimensional
  - Sorted map

Copyright Ellis Horowitz, 2011-2022

Bigtable Storage Model

- **Data is indexed using row and column names**
- **Data is treated as uninterpreted strings**
  - (row:string, column:string, time:int64) → string
- **Rows**
  - Data maintained in lexicographic order by row key
  - Tablet: rows with consecutive keys
    - Units of distribution and load balancing
- **Columns**
  - Column families
- **Cells**
- **Timestamps**

Rows

- Name is an arbitrary string
  - Access to data in a row is atomic
  - Row creation is implicit upon storing data
- Rows ordered lexicographically
  - Rows close together lexicographically usually reside on one or a small number of machines
- Each row/column intersection can contain multiple cells
  - Each cell contains a unique timestamped version of the data for that row and column
- Storing multiple cells in a column provides a record of how the stored data has changed over time

Copyright Ellis Horowitz, 2011-2022

Columns

- **Columns have two-level name structure:**
  - **family:optional_qualifier**
- **Column family**
  - **Unit of access control**
  - **Has associated type information**
- **Qualifier gives unbounded columns**
  - **Additional level of indexing, if desired**

Copyright Ellis Horowitz, 2011-2022

Timestamps

- Used to store different versions of data in a cell
  - New writes default to current time, but timestamps for writes can also be set explicitly by clients
- Garbage Collection
  - Per-column-family settings to tell Bigtable to GC
  - *"Only retain most recent K values in a cell"*
  - *"Keep values until they are older than K seconds"*
- API: Create / delete tables and column families

Copyright Ellis Horowitz, 2011-2022

Data Model – WebTable Example (1 of 7)

A large collection of web pages and related information

Copyright Ellis Horowitz, 2011-2022

Data Model – WebTable Example (2)

Tablet - Group of rows with consecutive keys.
Unit of Distribution

Bigtable maintains data in lexicographic order by row key

Copyright Ellis Horowitz, 2011-2022

Data Model – WebTable Example (3)

Column Family

Column family is the unit of access control

Copyright Ellis Horowitz, 2011-2022

Data Model – WebTable Example (4)

Column

Column key is specified by
"**Column family:qualifier**"

Copyright Ellis Horowitz, 2011-2022

Data Model – WebTable Example (5)

Column

You can add a column in a column family if the column family was created

Copyright Ellis Horowitz, 2011-2022

Data Model – WebTable Example (6)

Cell: the storage referenced by a particular **row key**, **column key**, and **timestamp**

Copyright Ellis Horowitz, 2011-2022

Data Model – WebTable Example (7)

Different cells in a table can contain multiple versions indexed by timestamp

Copyright Ellis Horowitz, 2011-2022

BigTable is Used on many Real Applications

Project name	Table size (TB)	Compression ratio	# Cells (billions)	# Column Families	# Locality Groups	% in memory	Latency-sensitive?
Crawl	800	11%	1000	16	8	0%	No
Crawl	50	33%	200	2	2	0%	No
Google Analytics	20	29%	10	1	1	0%	Yes
Google Analytics	200	14%	80	1	1	0%	Yes
Google Base	2	31%	10	29	3	15%	Yes
Google Earth	0.5	64%	8	7	2	33%	Yes
Google Earth	70	–	9	8	3	0%	No
Orkut	9	–	0.9	8	5	1%	Yes
Personalized Search	4	47%	6	93	11	5%	Yes

Copyright Ellis Horowitz, 2011-2022

References

- **Google Videos on map/reduce**
  https://www.youtube.com/watch?v=yjPBkvYh-ss (Lecture 1, 46 min)
  https://www.youtube.com/watch?v=-vD6PUdf3Js (Lecture 2, 52 min)
- **Wikipedia,** http://en.wikipedia.org/wiki/MapReduce
- *Data-Intensive Text Processing with MapReduce*, Jimmy Lin and Chris Dyer, Morgan & Claypool Synthesis Lectures on Human Language Technologies, 2010
  http://www.umiacs.umd.edu/~jimmylin/MapReduce-book-final.pdf
- **Hadoop** is an open source implementation of MapReduce
  http://hadoop.apache.org/
- MapReduce: Simplified Data Processing on Large Clusters, by Jeffrey Dean and Sanjay Ghemawat, http://research.google.com/archive/mapreduce.html

Copyright Ellis Horowitz, 2011-2022

Other Useful Videos

- **How does Google search work?**
  - **https://www.youtube.com/watch?v=KyCYyoGusqs** (7 min)
- **How does Google decide when to display multiple results from the same website**
  - **https://www.youtube.com/watch?v=AGpEdyIcZcU** (6 min)*
- **Larry Page of the future directions of Google**
  - **http://www.youtube.com/watch?v=mArrNRWQEso** (3/2014, 23 min)
- **How Search Works by Matt Cutts**
  - **http://www.youtube.com/watch?v=BNHR6IQJGZs** (3 min)

Copyright Ellis Horowitz, 2011-2022

The Search Engine Business Model Advertising

Copyright Ellis Horowitz 2011-2022

Types of Online Advertising

- **Banner Advertising**
  - The earliest form of online ads
  - People have developed "banner blindness"
- **Pay-per-click Advertising**
  - Introduced by search engines Yahoo/Google/Bing/etc
- **Website Advertising**
  - Place ads on blogs/newsletters/websites that typically attract the customers you are aiming at
- **Affiliate Marketing/build a platform and they will come**
  - Let third party sites advertise your goods/services and give them a commission
  - Amazon (https://affiliate-program.amazon.com/)
  - Ebay (http://pages.ebay.com/affiliate/referral.html)
- **Social Media Marketing**
  - The use of social media platforms (e.g. Facebook, Instagram, TikTok) and websites to promote a product or service

Copyright Ellis Horowitz, 2011-2022

How Big is Search Engine Advertising

Google/Alphabet earned more than $257 billion in 2021

Alibaba earned $109 billion in 2021

Yahoo earned $4.6 billion in 2019
(Verizon has purchased Yahoo for $4.5 billion in 2015)

Bing, a division of Microsoft earned $7.7 billion in 2021

Baidu earned $5.9 billion in 2021

Google and Baidu Stock History

Snapshots 10/2021

Copyright Ellis Horowitz 2011-2021

Positional Bias is Very Strong

A google result page showing where people looked and where people clicked

Eye pupil hotspots

See **http://www.entertop.net/**

X marks the clicks

Copyright Horowitz 2007-2022

Estimating the Value of Rank in a Sponsored List

- A survey worked by examining click data from AOL, HitWise and Overture yielding statistics about how often a user clicked on the first search result, second search result, etc.
- Total Searches: 9,038,794
- Total Clicks: 4,926,623
- Ranking Number 1 receives 42.1 percent of click throughs.
  Ranking Number 2 receives 11.9 percent of click throughs.
  Ranking Number 3 receives 8.5 percent of click throughs.
  Ranking Number 4 receives 6.1 percent of click throughs.
  Ranking Number 5 receives 4.9 percent of click throughs.
  Ranking Number 6 receives 4.1 percent of click throughs.
  Ranking Number 7 receives 3.4 percent of click throughs.
  Ranking Number 8 receives 3.0 percent of click throughs.
  Ranking Number 9 receives 2.8 percent of click throughs.
- Ranking Number 10 receives 3.0 percent of click throughs.

Copyright Ellis Horowitz, 2011-2022

THE MILLION DOLLAR QUESTION

Q. How do I get to number one in Google search results?

Google's organic search results are produced by a proprietary algorithm that is often being changed, so there are never any guarantees that someone will continue to be highly ranked

But there are lots of people ready to give you advice

Copyright Horowitz 2007-2022

Search Engine Optimization

- **There is an industry of companies whose focus is to develop and refine a company's online presence;**
  - They are called Search Engine Optimizers
- **Search Engine Optimization (SEO) involves:**
  - Making pages show up higher in search engine's organic results
  - Optimizing content to target certain keyword phrases
  - Developing web page content that responds to each seeker's interests
- **There are many companies who claim to do SEO**
  - For a list see http://www.topseos.com/rankings-of-best-seo-companies
  - Generally they are "small" companies, earning under $10 million per year

Recent Google horizontal card-style view

THE MILLION DOLLAR ANSWER

Q. How do I get to number one in Google?
A. Use paid search

Paid search is one answer; Google and many other search engines use the pay-per-click (PPC) model of advertising,

Pay enough for a click and you can pretty much guarantee to get a top spot on the list of ads for your chosen keywords

Lets Take a Close Look at Google Adwords

Renamed as Google Ads
Google's program for accepting pay-per-click ads.
Its home page is https://ads.google.com

For every keyword phrase there is an auction where bidders agree to pay a certain amount to Google if their ad is clicked on;

Copyright Ellis Horowitz, 2011-2022

Campaigns Begin By Choosing Keywords Google Keyword Estimator

For specific keywords Google provides the approximate cost per click and the resulting position in the list, and other stats

Maximum CPC: Daily budget: Get New Estimates

Keywords ▼	Search Volume	Estimated Avg. CPC	Estimated Ad Positions	Estimated Clicks / Day	Estimated Cost / Day
Search Total		**$5.23 - $7.34**	**1 - 3**	**270 - 348**	**$1,450 - $2,660**
crime scene detective class		Not enough data to give estimates. [?]			
criminal justice		$4.85 - $6.84	1 - 3	221 - 277	$1,080 - $1,900
criminal justice associates degree		$9.58 - $12.76	1 - 3	0 - 1	$0 - $20
criminal justice career		$3.53 - $4.41	1 - 3	2 - 3	$8 - $20
criminal justice career training		Not enough data to give estimates. [?]			
criminal justice classes		$8.72 - $10.90	1 - 3	0 - 1	$0 - $20
criminal justice college		$4.76 - $5.95	1 - 3	10 - 14	$50 - $90
criminal justice course		$10.63 - $15.94	1 - 3	0 - 1	$0 - $20
criminal justice coursework		Not enough data to give estimates. [?]			
criminal justice degree		$13.76 - $20.64	1 - 3	16 - 21	$230 - $440
criminal justice online class		Not enough data to give estimates. [?]			
criminal justice program		$7.34 - $9.92	1 - 3	2 - 4	$20 - $40
criminal justice school		$7.37 - $9.31	1 - 3	4 - 6	$30 - $60
detective training		$3.58 - $4.48	1 - 3	2 - 3	$8 - $20
law enforcement career		$4.40 - $5.49	1 - 3	3 - 4	$20 - $30
law enforcement program		$3.29 - $4.12	1 - 3	0 - 1	$0 - $5
law enforcement training		$3.44 - $4.30	1 - 3	10 - 12	$40 - $60

Estimates for these keywords are based on clickthrough rates for current advertisers. Some of the keywords above are subject to review by Google and may not trigger your ads until they are approved. Please note that your traffic estimates assume your keywords are approved.

Long Tailed Keywords can be Valuable

- Long-tail keywords are search queries made up of three-four word phrases that are very specific to a product, good, or service that's being sold.
  - Long-tail keywords are the phrases search engine users are generally more likely to type in when they're closer to purchasing an item.

- The conversion rate for long-tail keywords is approximately **2.5 times higher than it is for head (shorter) keywords.**
- long-tail keywords that present *less* competition also offer *lower cost-per-click prices* since few marketers are targeting them

Copyright Ellis Horowitz, 2011-2022

Your Keyword Phrases Should be Mapped on Your Website

- **Use key phrases in the content on your page**
- **Develop metadata that includes key phrases in:**
  - TITLE tags
  - Meta Description and Keyword tags
  - ALT tags
- **Name directories, files and images with the same key words or phrases**

Notice how folder names mimic the data they contain

- **Query:** `"Levi's men's clothing jeans"`
- produces the URL
- **https://www.levi.com/US/en_US/clothing/men/jeans/c/levi_clothing_men_jeans/**
- **Query:** `"Ladies handbags"`
- **https://www.neimanmarcus.com/c/handbags-all-handbags-evening-bags-cat35400731/**

Copyright Ellis Horowitz, 2011-2022

Advertisers Designate Keyword Matching Rules

- The *advertiser* will specify the type of matching to be done against his keyword phrases
  - Four types of keyword matching options can help you refine your ad targeting

    1. Broad Match
    2. Exact Match
    3. Phrase Match
    4. Negative Keyword

Copyright Ellis Horowitz, 2011-2022

Keyword: Broad Match

- **A broad match is the default option**
  - When you include keyword phrases like "tennis shoes" in your keyword list, your ads will appear when users search for "tennis" and "shoes", in any order (and possibly along with other terms)
- **Broad Match ads may also show for expanded matches, including synonyms and plurals**
  - Google will analyze your keyword list, ad text, and daily Google search queries, and show your ads for relevant terms and variations (such as "tennis sneakers")
    - Even if you didn't include these terms in your keyword list
- **Broad matches are often less targeted than exact or phrase matches**
  - If you decide to run your ads on broad-matched keywords, create keyword phrases containing at least two descriptive words

https://support.google.com/google-ads/answer/2497828

Copyright Ellis Horowitz, 2011-2022

Keyword: Exact Match

- **The search query must exactly match your keyword**
  - Originally Exact Match meant that "tennis shoes" would only match a user request for "tennis shoes" and not for "red tennis shoes," even though the second query contains your keywords
  - However, Google now includes rewording and reordering for exact match keywords
  - Exact match will now ignore function words (in, to), conjunctions (for, but), articles (a, the) and other words that don't impact the intent of the query
  - https://support.google.com/google-ads/answer/7478529

Keyword	Query	Why it matched
jobs in united states nike shoes women	jobs in **the** united states nike shoes **for** women	**Function words added**
parks **in** san diego paint for **a** deck	parks san diego paint for deck	**Function words removed**
news **from** today bahamas cruise **from** miami	news **for** today miami **to** bahamas cruise	**Function words changed**

Keyword	Query	Why it matched
running shoes men's dress shirt	shoes running dress shirt men's	**Words reordered**

https://support.google.com/google-ads/answer/2497825

Copyright Ellis Horowitz, 2011-2022

Keyword: Phrase Match

- **Your ad appears when users search on the exact phrase**
  - AND when their search contains additional terms
    - As long as the keyword phrase is in exactly the same order
  - A phrase match for "tennis shoes" would include "red tennis shoes" but not "shoes for tennis"

https://support.google.com/google-ads/answer/2407784

Copyright Ellis Horowitz, 2011-2022

Keyword: Negative Keyword

- **Negative keywords allow you to eliminate searches that you know are not related to your message**
  - **If you add the negative keyword "–table" to your keyword "tennis shoes," your ad will not appear when a user searches on "table tennis shoes“**
    - Negative keywords should be used with caution, -- they can eliminate a large portion of a desired audience if applied incorrectly

https://support.google.com/google-ads/answer/2453972

Copyright Ellis Horowitz, 2011-2022

A Sample Google Ads Campaign Screen

Campaign Name	Current Status	Current Budget [?]	Clicks	Impr.	CTR	Avg. CPC	Cost
Campaign #2	Deleted	[$10.00 / day]	747	87,551	0.85%	$0.44	$326.34
Campaign #1	Deleted	[$5.00 / day]	328	9,893	3.31%	$0.53	$174.94
Total - all 2 campaigns	-	**$0.00 / day** active campaigns	**1,075**	**97,444**	**1.10%**	**$0.47**	**$501.28**

- **Ads advertisers create advertising campaigns**
  - There are two campaigns defined above
- **Each campaign has a set of keyword and associated ads, and includes**
  - a budget, recorded clicks, recorded impressions, click thru rate, average cost/click, and total cost of the campaign

Copyright Ellis Horowitz, 2011-2022

Capabilities of Search Engine Ad Servers

- **The typical common functionality of ad servers includes:**
  - Uploading the creative (*creative* is the term used to describe the ad that will appear)
  - Maintaining business rules for placing ads
  - Targeting ads to different users or content
  - Optimizing appearance of a set of creatives based upon results (choosing the most effective ad)
  - Reporting impressions, clicks, post-click activities, and interaction metrics.
- **Advanced functionality may include:**
  - *Frequency capping* creatives so users only see messages a limited amount of time.
  - *Sequencing* creatives so users see messages in a specific order (sometimes known as surround sessions.
  - Excluding competitive creatives so users do not see competitors' ads directly next to one another.
  - Displaying creatives so an advertiser can own 100% of the inventory on a page (sometimes known as roadblocks).
  - Targeting creatives to users based on their previous behavior (behavioral marketing or behavioral targeting).

Copyright Ellis Horowitz, 2011-2022

Search Engine Ad Servers are Versatile

- **Advertisers with accounts on Google's Ads can define a set of criteria for placement of their ad; typical criteria might include rules such as**
  - Only display my ad
    - from 9:00AM-5:00PM EST
    - once/day
    - if the viewer is located in the United States
  - *dayparting* is a technique that involves increasing your bids during times when conversion rates are typically above average, and decreasing them when rates are typically below average

Google Ads Auction Rules

- **Each bidder specifies (i) search terms that trigger its bid and (ii) the amount to bid for each search term.**
  - Bidders may also establish an overall ad budget and limits for each kind of bid.
  - Google may set a reserve or minimum price for each term.
- **Google estimates the "click-through rate" that each bidder would have if it were listed in the first spot.**
  - Ads with very low click-through rates are not displayed.
  - Google ranks bids by multiplying the click-through-rate and the bid amount.
  - Ads are displayed in rank order.
- **Google is paid only when an ad gets clicked.**
  - In that case, the price it receives is *the smallest price the bidder could have bid to get its ranking*.

Copyright Ellis Horowitz, 2011-2022

How does AdWords bidding work?

- The actual position of your ad is determined by your ad rank (Maximum Bid multiplied by the Quality Score).
- The highest ad rank gets the 1st ad position.
- Your actual CPC (cost per click) will be determined by the ad rank of the next highest ad below you
- **Exception**: when you are the only bidder or the lowest bid in the Ads auction; then you pay your maximum bid per click!
- Ads bidding heavily penalizes advertisers who bid with low quality scores. Conversely, those with high Quality Scores get higher ad ranks and lower CPC
- The average cost per click on Ads varies by keyword and industry, but is roughly $2.00 on the search network and $0.58 on the display network

Copyright Ellis Horowitz, 2011-2022

Some Facts About Online Advertising

- **https://www.wordstream.com/blog/ws/2016/02/29/google-adwords-industry-benchmarks**
- **A study of over 14,000 US based clients in multiple markets**
- **The average *click-through rate* in Ads across all industries is 3.17% for search and 0.46% for display ads**
  - How often people click on your ad versus how often it is displayed
- **The average *cost per click* in Ads across all industries is $2.69 for search and $0.63 for display**
- **The average *conversion rate* in Ads across all industries is 3.75% for search and 0.77% for display**
  - How often a click on an ad leads to a sale
- **The average *Cost Per Action* (CPA) in Ads across all industries is $48.96 for search and $75.51 for display**
  - Refers to a specific action such as a sale, click, or form submission

GDN refers to the Google Display Network where ads are placed on 3rd party websites

Copyright Ellis Horowitz, 2011-2022

Average Click Thru Rate by Industry

Bars in order:

- Advocacy
- Auto
- B2B
- Consumer services
- Dating and personals
- E-commerce
- Education
- Employment services
- Finance & insurance
- Health & medical
- Home goods
- Industrial services
- Legal
- Real estate
- Technology
- Travel & hospitality

Second Price Auction Using Only Bid Amount

- In a **First-Price Auction** if your bid wins, you pay exactly what you bid. This maximizes revenue potential for the seller.
- **A Second-Price Auction** is a digital buying model where if your bid wins, you pay $0.01 above the **second** highest bid in the **auction**.
  - In this type of **auction**, it is in your best interest to bid the highest amount you are willing to pay, knowing that often you will end up paying less than that amount.

Bid2: ($9)

Bid3: ($6)

Bid1: ($5)

- **Bidder 2 is ranked 1st**
  - **Pays $6+1¢=$6.01**
- **Bidder 3 is ranked 2nd**
  - **Pays $5+1¢=$5.01**

Let P(C) be the Probability of the Ad Being Clicked

Bid2:	*$9*	*P(C)=0.1*	*$0.9*
Bid3:	*$6*	*P(C)=0.1*	*$0.6*
Bid1:	*$5*	*P(C)=0.4*	*$2.0*

Expected Revenue

P(c) can depend upon:

- Historical click performance of the ad
- Landing page quality
- Relevance to the user
- User click through rates

Expected Revenue = Bid amount X Click Probability
So the ad with highest bid does not necessarily produce the most revenue

Google Ranks Ads by the Product of Bid Amount*Click Probability

Bid1:	*$5*	*P(C)=0.4*	*is placed 1st*
Bid2:	*$9*	*P(C)=0.1*	*is placed 2nd*
Bid3:	*$6*	*P(C)=0.1*	*is placed 3rd*

Ad Rank= Bid X Click Probability

Return On Investment: What is Each Conversion Worth?

Copyright Ellis Horowitz, 2011-2022

Improved Conversion Rate

Copyright Ellis Horowitz, 2011-2022

AdSense

- **AdSense** from Google is a service for placing Google ads on third party web pages
- Google purchased the content matching technology from Oingo, a small Santa Monica-based search engine in 2003 for $102 million
- The program is designed for website publishers who want to display targeted text, video or image advertisements on their website pages and earn money when site visitors view or click the ads
- Typically Google keeps 32% of the ad revenue and returns 68% to the website developer
- In addition to cost-per-click ads, Google also offers
  - *Cost per Thousand* displays, CPM
  - *Cost per Engagement*, where advertisers only pay when users actively engage with ads, e.g. hovering over them or expanding them or reviewing the landing page for more than a few seconds
- **Typically blogs use AdSense to monetize their site**

Copyright Ellis Horowitz, 2011-2022

How AdSense Works

1. The webmaster inserts the AdSense JavaScript code into a webpage.
2. Each time this page is visited, the JavaScript code uses inlined JSON to display content fetched from Google's servers.
3. Google's servers use a cache of the page to determine a set of high-value keywords. Ads are served for those keywords based on the AdWords bidding system.
4. For site-targeted ads, the advertiser chooses the page(s) on which to display ads, and pays based on CPM, or the price advertisers choose to pay for every thousand advertisements displayed.
5. Search ads are added to the list of results after the visitor performs a search.
6. To protect against fraud, AdSense customers can specify the pages on which ads should be shown. AdSense then ignores clicks from pages other than those specified.

Copyright Ellis Horowitz, 2011-2022

AdSense Ads on Websites are Not Always Relevant

Google's **AdSense** program claims to place ads on third-party websites, where the ads are **relevant** to the site's content;

Each time a visitor visits a page with an AdSense tag, a piece of JavaScript writes an iframe tag, whose src attribute includes the URL of the page. Google's servers use a cache of the page to determine a set of high-value keywords.

Above, an article about Michael Jackson includes an irrelevant ad about payday loans

Copyright Ellis Horowitz, 2011-2022

Adsense Content Matching Originally Based on WordNet

- **AdSense technology is based upon a database of word meanings initially developed at Princeton, called WordNet**
  - https://wordnet.princeton.edu/
- **WordNet is a semantic lexicon for the English language. It groups English words into sets of synonyms called *synsets*, provides short, general definitions, and records the various semantic relations between these synonym sets.**
  - The purpose is twofold:

  1. to produce a combination of dictionary and thesaurus that is more intuitively usable, and
  2. to support automatic text analysis.

Copyright Ellis Horowitz, 2011-2022

AdSense Code

- Google AdSense creates the code, which the publisher copies and pastes into a web page or blog post at the point where it is to appear.
- This is an example of AdSense code for a 728-by-90-pixel ad. Google assigns the numbers following "google_ad_client" and "google_ad_slot."
-
```
<script type="text/javascript"><!--
google_ad_client = "xxxxxxxxxxxxxxxx";
/* 728x90, created 10/28/08 */
google_ad_slot = "xxxxxxxxxx";
google_ad_width = 728;
google_ad_height = 90;
//-->
</script>
<script type="text/javascript"
src="http://pagead2.googlesyndication.com/pagead/show_ads.js">
</script>
```
- When a webpage containing this code is displayed, AdSense counts the page visit. If the visitor clicks on the ad, s/he is redirected to the advertiser's website, and Google counts the click and credits the AdSense publisher's account
- Payments for clicks on ads range from a few cents to several dollars.

Copyright Ellis Horowitz, 2011-2022

Sample Google Conversion Code for Tracking Purchases/Sales

- Google/Yahoo offer tracking pixels
- Tracking pixels are small, typically transparent images on a web page that have special names which permit the loading of the web page to be tracked by a web server.

```
<!-- Google Code for purchase Conversion Page -->
<script language="JavaScript" type="text/javascript">
var google_conversion_id = 1072192880;
var google_conversion_language = "en_US";
var google_conversion_format = "1";
var google_conversion_color = "666666";
if (1) { var google_conversion_value = 1; }
var google_conversion_label = "purchase";
</script>
<script language="JavaScript"
   src="http://www.googleadservices.com/pagead/conversion.js"> </script>
<noscript>
<img height=1 width=1 border=0
   src="http://www.googleadservices.com/pagead/conversion/1072192880/?value
   =1&label=purchase&script=0"> </noscript>
```

The code is inserted into a page; an external Javascript file is imported; A small image, a pixel, is placed on the page; When the page is requested, Google records the access

Copyright Ellis Horowitz, 2011-2022

Google AdSense, AdMob Moves to a First-Price Auction Model

- **First-price vs. second-price auctions**
  - In a second-price auction, the final price paid by the winner is determined by the second-highest bid.
  - In a first-price auction, the final price is the same as the winning bid.
- ***This change applies to AdSense, Ad Manager and AdMob***
- AdMob is a mobile advertising company founded by Omar Hamoui. The name AdMob is a portmanteau for "advertising on mobile"
- Showing ads to app users allows app developers to create a sustainable source of revenue
- Ads are created and paid for by advertisers who want to promote their products or services to app users. Once you create space for ads in your app, AdMob works with advertisers who pay to show ads that are relevant to your users

Copyright Ellis Horowitz, 2011-2022

Ad Exchanges and DoubleClick

- An **ad exchange** is a technology platform that facilitates the buying and selling of media advertising inventory from multiple **ad** networks.
- Prices for the inventory are determined through technology-driven bidding
- The key function of an **ad network** is aggregation of **ad** space supply from publishers and matching it with advertiser demand
- DoubleClick **Ad Exchange** is a real-time marketplace owned by **Google** for buying and selling advertising
  - *In 2018 Google changed the name of DoubleClick to Google Ad Manager*

Watch "The Social Dilemma" documentary on NetFlix
https://www.netflix.com/title/81254224

Copyright Ellis Horowitz, 2011-2022

The Real-Time Bidding Paradigm in Online Advertising

People browse the internet and navigate to a content-based website

The website makes money by showing ads

Some ads are sold direct to major advertisers

But unsold ad space is liquidated to online advertising exchanges

Exchanges turn around and offer these impressions to real time bidding (RTB) platforms

Each impression is put out for bid to a bank of RTB platforms

RTB platforms evaluate the value of that specific impression across the campaigns it is executing. The winning bidder serves the ad to the site visitor

© 2010 Hurdle.biz

* All associations between advertisers and other organizations are only meant to be an example and do not represent real relationships

Today's advertising model

Trackers

Publishers

Advertisers

Broker (Ad exchange)

Client

The cast of players

Copyright Ellis Horowitz 2011-2022

Copyright Ellis Horowitz 2011-2022

Trackers may share profiles with advertisers?

Copyright Ellis Horowitz 2011-2022

Copyright Ellis Horowitz 2011-2022

Copyright Ellis Horowitz 2011-2022

Broker launches auction (for given user visiting given webpage ….)
Also does clickfraud etc.

Copyright Ellis Horowitz 2011-2022

Copyright Ellis Horowitz 2011-2022

Copyright Ellis Horowitz 2011-2022

Copyright Ellis Horowitz 2011-2022

Various reporting of results . . . .

Copyright Ellis Horowitz 2011-2022

How an Advertising Network such as DoubleClick Works

Copyright Ellis Horowitz, 2011-2022

Search Engines and the Use of Knowledge-Based Systems

Copyright Ellis Horowitz, 2011-2022

Outline

- Basic definitions: Taxonomy, Ontology, Knowledgebase
- Knowledgebase Internals and Examples
- WordNet
- Wikipedia
- Google's Knowledge Graph

Copyright Ellis Horowitz, 2011-2022

What is a Taxonomy

What is a taxonomy

ταξις, ***taxis***
"arrangement"

+

νομος, ***nomos***
"law"

A taxonomy is a classification or categorization of a complex system.

Football team

is a

Real Madrid C.F.

e.g. **the ACM Computing Classification System, https://dl.acm.org/ccs**
the Mathematics Subject Classification

What is an Ontology

- a set of concepts and categories in a subject area or domain that shows their properties and the relations between them, or
- a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents, or
- a body of formally represented knowledge based on a *conceptualization*: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them

Brief ontology of movies

Copyright Ellis Horowitz, 2011-2022

What is a Knowledgebase

- A **knowledgebase** (**KB**) is a technology used to store and retrieve complex structured and unstructured information as stored in an ontology
  - Two components:
  - 1. a way of ***representing*** information and
  - 2. a method for ***reasoning*** about the information
- A **knowledgebase** is a collection of information stored in an ontology that includes software used to author and present that information

Copyright Ellis Horowitz, 2011-2022

Is Elvis Alive

Ask.com – Mostly Yes

Bing.com – Yes/Maybe

Google.com - No

Wolfram Alpha - No

2 Yes'es and 2 No's;

text matching alone is insufficient

we need to "understand" the query

Copyright Ellis Horowitz 2011-2022

KnowledgeBases-Developed by AI Community

- **To move away from just using keyword matching, search engines borrowed techniques developed by AI researchers**
- A ***knowledgebase*** (KB) is a technology used to store complex structured and unstructured information used by a computer system.
- **A knowledge-based system consists of two elements:**
  1. a *knowledgebase* that ***represents*** facts about the world, and
  2. an *inference engine* that can ***reason*** about those facts (and use rules and other forms of logic to deduce new facts or highlight inconsistencies)
- **The term "knowledgebase" was coined to distinguish this form of knowledge store from the more common and widely used term *database***
  - Is a relational database a knowledgebase?
- ***Examples of knowledgebases: Freebase, Google's Knowledge Graph, Apple's Siri, IBM's Watson***

Copyright Ellis Horowitz, 2011-2022

Search Engines Use Knowledgebases to Enhance the Display of Results

- **The representation of knowledge in a knowledgebase is an *object model***
  - Includes classes, subclasses and instances
- A **taxonomy** is usually only a hierarchy of concepts (i.e. the *only relation* between the concepts is parent/child, or subClass/superClass, or broader/narrower)
- In a **knowledgebase**, *arbitrary complex relations* between concepts can be expressed as well, e.g. (*X marriedTo Y*; or *A worksFor B*; or *C locatedIn D*, etc)
- **Search engines utilize this linked, structured data in many ways, such as**
  - Providing direct answers to queries
  - enhanced displays in many varieties of engaging visual formats, e.g. see query "Picasso" in Google

Copyright Ellis Horowitz, 2011-2022

Close Up: A KnowledgeBase for Elvis

Information Extraction

Content analytics

"Elvis Presley, the first and greatest American rock-and roll star, died yesterday at the age of 42."

Types of Knowledge For a KnowledgeBase

Elvis Presley type American singer Elvis Presley type Baritone American singer subclassOf singer	taxonomic knowledge
Elvis Presley sang All Shook Up Elvis Presley bornIn Tupelo	factual knowledge
id11: Elvis Presley marriedTo Priscilla Presley id11 validDuring [1967, 1977]	temporal knowledge
Elvis Presley „has twin brother“ Jesse Garon Elvis Presley „possibly has origin“ Cherokee	emerging knowledge
Elvis Presley knownAs „The King of R&R“	terminological knowledge

Taxonomies are narrower than ontologies since ontologies include a larger variety of relation types. Mathematically, a hierarchical **taxonomy** is a tree structure of classifications for a given set of objects An ontology is a directed, labeled, cyclic graph.

Copyright Ellis Horowitz 2011-2022

Properties of a Good Knowledge Representation System

- **Whatever knowledge representation scheme has been chosen to represent a particular domain it should exhibit the following four properties**
- ***Representation adequacy***
  - **The ability to represent all kinds to knowledge that are needed in the domain**
- ***Inferential ability***
  - **The ability to manipulate the representational structures in such a way as to derive new structures by the process of differencing**
- ***Inferential efficiency***
  - **The ability to incorporate into the knowledge structure additional information that can be used to guide the inference mechanism towards the most promising path**
- ***Acquisitional efficiency***
  - **The ability to acquire new information easily, i.e. to automatically update the structure with new knowledge**

Copyright Ellis Horowitz, 2011-2022

Inheritance

- An important feature of knowledge representation is its organization into *class hierarchies*. Classes can be based on the properties of objects/concepts
- Inheritance is a relation by which
  - 1. an individual assumes the properties of its class and
  - 2. determines which properties of a class are passed on to its subclass
- In order to support inheritance, objects/concepts must be organized into classes and classes must be arranged in a generalized hierarchy
  - the most generic object/concept is at the top of the hierarchy and the most specific is at the bottom of the hierarchy

Copyright Ellis Horowitz, 2011-2022

Inheritance

Copyright Ellis Horowitz, 2011-2022

Some Advantages of Inheritance

- Inheritance provides a natural mechanism for *representing taxonomically structured knowledge*
- Inheritance provides an economical means of *expressing properties common to a class of objects/concepts*
- Inheritance *guarantees that all members of a class inherit the appropriate properties* thus ensuring consistency with the class definition
- Inheritance *reduces the size of the knowledge base* as properties are defined once for the most general type of object/concepts and these properties are then share by other less generic objects/concepts in the type hierarchy.

Copyright Ellis Horowitz, 2011-2022

Different Notations for a KnowledgeBase

- Resource Description Format (RDF) is a W3C spec used for creating ontologies;
  - https://www.w3.org/RDF/
  - Sometimes "RDF Ontology" and "KnowledgeBase (KB)" are used synonymously.

Graph notation:

Triple notation:

Subject	Predicate	Object
Elvis	type	singer
Elvis	bornIn	Tupelo
…	…	…

Logical notation:

type(Elvis, singer)
bornIn(Elvis,Tupelo)
…

RDF Data Model

- RDF allows us to make statements about resources.
  The format of these statements is:
  <subject> <predicate> <object>
- Some examples

<Bob> <is a> <person>
<Bob> <is a friend of> <Alice>
<Bob> <is born on> <the 4th of July 1990>
<Bob> <is interested in> <the Mona Lisa>
<the Mona Lisa> <was created by><Leonardo da Vinci>
<the video 'La Joconde à Washington'> <is about> <the Mona Lisa>

We can visualize triples as a connected **graph**.
Graphs consists of nodes and arcs.
The subjects and objects of the triples make up the nodes in the graph;
the predicates form the arcs

One query language for making inferences on these graphs is SPARQL

To Answer a Question Knowledgebases Need to Combine Information From *Multiple Sources*

Copyright Ellis Horowitz, 2011-2022

KnowledgeBases are Can Be Represented as Labeled MultiGraphs

A knowledgebase can be seen as a directed labeled multigraph, where the nodes are entities and the edges relations.

A **multigraph** is a graph which is permitted to have multiple edges that have the same end nodes. Two vertices may be connected by more than one edge

A Single Entity Can Have Different Labels

To Build a Knowledgebase One Must Find Classes and Instances

Which classes exist? (aka entity types, unary predicates, concepts)

Which subsumptions hold?

Which entities belong to which classes?

Which entities exist?

Two Knowledgebases With Complementary Information

See https://en.wikipedia.org/wiki/YAGO_(database)

See https://theelvispedia.com/

A Knowledgebase Must Work Across Multiple Ontologies

Who is the spouse of the guitar player?

YAGO

ElvisPedia

We Need to Match Entities, Classes and Relations

Combining Elements From Different Knowledgebases Means Matching Entities

Build a graph:

nodes: pairs of entities, weighted with similarity

edges: weighted with degree of relatedness

similarity: 0.9

similarity: 0.7

similarity: 0.8

Iterate until convergence:

similarity := weighted sum of neighbor similarities

many variants (belief propagation, label propagation, etc.)

Inferencing on KnowledgeBases

- An **inference engine** is a component of a system that applies logical rules to a knowledgebase to deduce new information
- This process is ongoing as each new fact in the knowledgebase can trigger additional rules in the inference engine.
- Inference engines work primarily in one of two modes: forward chaining and backward chaining
  - **Forward chaining** starts with the known facts and asserts new facts.
  - **Backward chaining** starts with goals, and works backward to determine what facts must be asserted so that the goals can be achieved.
- An inference engine cycles through three sequential steps: match rules, select rules, and execute rules
- The execution of the rules will sometimes result in new facts or goals being added to the knowledgebase which will trigger the cycle to repeat. This cycle continues until no new rules can be matched
- Search engines typically use forward chaining

Copyright Ellis Horowitz, 2011-2022

Forward Chaining

- **Forward chaining** is the repeated application of modus ponens
- In propositional logic, ***modus ponens*** is the rule
  - "*P* implies *Q*" and "*P*" are both asserted to be true, so therefore *Q* must be true."
  - Sometimes modus ponens is written as ((P -> Q) and P) -> Q
  - For Example
    - **If today is Tuesday, then John will go to work.**
    - **Today is Tuesday.**
    - **Therefore, John will go to work.**
- Forward chaining starts with the available data and uses inference rules to extract more data until a goal or endpoint is reached.

Copyright Ellis Horowitz, 2011-2022

Binary Relations and Instances

Here are some sample binary relations with their type signature, e.g.

hasAdvisor: Person × Person
graduatedAt: Person × University
bornOn: Person × Date

Here are instances of the above binary relations

hasAdvisor (JimGray, MikeHarrison)
hasAdvisor (Susan Davidson, Hector Garcia-Molina)
graduatedAt (JimGray, Berkeley)
graduatedAt (HectorGarcia-Molina, Stanford)
bornOn (JohnLennon, 9-Oct-1940)

For more examples see
https://www.tutorialandexample.com/forward-chaining/
And
https://www.javatpoint.com/forward-chaining-and-backward-chaining-in-ai

One Approach to Inferencing

A small portion of a knowledgebase with entities Novel, Jane Eyre, etc. and relations HasFather, Wrote, IsA and inverse relations IsA^{-1}

a **Horn clause** is a logical formula whose form is useful in logic programming,
A Horn clause is a disjunction of literals with at most one positive, i.e. unnegated literal

HasFather(a, b) ^ isa(b,y) ➔ isa(a; y)
Write(a, i) ^ isa(i, x) ^ isa(j,x) ^ Write(b, j) ^ isa(b,y) ➔ isa(a; y)
InSentence(a, j) ^ InSentence(b, j) ^ isa(b,y) ➔ isa(a; y)

HasFather(x, a) ^ isa(a,writer) ➔ isa(x; writer)

Copyright Ellis Horowitz 2011-2022

Semantic Network

- A semantic network is a knowledge representation scheme that represents knowledge as a graph.
  - The nodes of the graph correspond to concepts or facts and
  - the arc correspond to relations or associations between concepts.
  - In a semantic network both the nodes and arcs are labeled

Copyright Ellis Horowitz, 2011-2022

Semantic Network

- A semantic network that defines the properties of snow and ice
- The concept snowman inherits all the properties of snow
- The concept of ice and snow share a number of properties

Expressing Relationship in Semantic Networks

- The most important link in a semantic network is the is-a link
- A semantic network can organize knowledge in a hierarchy by using the is-a link such that the lower nodes inherit the properties of the higher nodes

The is-a relationship in a Semantic Network

Copyright Ellis Horowitz, 2011-2022

WordNet

Copyright Ellis Horowitz 2011-2022

WordNet is a Lexical DataBase with many Classes, Subclasses, and Superclasses

WordNet contains 82,000 classes

WordNet contains thousands of subclassOf relationships

WordNet contains 118,000 class labels

- **WordNet,** developed at Princeton, is a lexical database for the English language.
  - It groups English words into sets of synonyms called **synsets**,
  - provides short definitions and usage examples,
  - records a number of relations among these synonym sets or their members.

Lexical means text-only

WordNet Example: Superclass of Person

- S: (n) **person**, individual, someone, somebody, mortal, soul (a human being) *"there was too much for one person to do"*
  - *direct hyponym* / *full hyponym*
  - *part meronym*
  - *member holonym*
  - *direct hypernym* / ***inherited hypernym*** / *sister term*
    - S: (n) organism, being (a living thing that has (or can develop) the ability to act or function independently)
      - S: (n) living thing, animate thing (a living (or once living) entity)
        - S: (n) whole, unit (an assemblage of parts that is regarded as a single entity) *"how big is that part compared to the whole?"; "the team is a unit"*
          - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow) *"it was full of rackets, balls and other objects"*
            - S: (n) physical entity (an entity that has physical existence)
              - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Note the terms:

- **hyponym**
  - More specific
- **Holonym**
  - Denoting the whole
- **Hypernym**
  - A broad or superordinate

WordNet Example: Subclass of Singer

- S: (n) **singer**, vocalist, vocalizer, vocaliser (a person who sings)
  - ***direct hyponym*** / *full hyponym*
    - S: (n) alto (a singer whose voice lies in the alto clef)
    - S: (n) baritone, barytone (a male singer)
    - S: (n) bass, basso (an adult male singer with the lowest voice)
    - S: (n) canary (a female singer)
    - S: (n) caroler, caroller (a singer of carols)
    - S: (n) castrato (a male singer who was castrated before puberty and retains a soprano or alto voice)
    - S: (n) chorister (a singer in a choir)
    - S: (n) contralto (a woman singer having a contralto voice)
    - S: (n) crooner, balladeer (a singer of popular ballads)
    - S: (n) folk singer, jongleur, minstrel, poet-singer, troubadour (a singer of folk songs)
    - S: (n) hummer (a singer who produces a tune without opening the lips or forming words)
    - S: (n) lieder singer (a singer of lieder)
    - S: (n) madrigalist (a singer of madrigals)
    - S: (n) opera star, operatic star (singer of lead role in an opera)
    - S: (n) rapper (someone who performs rap music)
    - S: (n) rock star (a famous singer of rock music)
    - S: (n) songster (a person who sings)
    - S: (n) soprano (a female singer)
    - S: (n) tenor (an adult male with a tenor voice)
    - S: (n) thrush (a woman who sings popular songs)
    - S: (n) torch singer (a singer (usually a woman) who specializes in singing torch songs)
    - S: (n) voice ((metonymy) a singer) *"he wanted to hear trained voices sing it"*
    - S: (n) warbler (a singer; usually a singer who adds embellishments to the song)

WordNet Example: Instances But Very Few

- S: (n) **singer**, vocalist, vocalizer, vocaliser (a person who sings)
  - *direct hyponym* / *full hyponym*
  - ***has instance***
    - S: (n) Bailey, Pearl Bailey, Pearl Mae Bailey (United States singer (1918-1990))
    - S: (n) Cash, Johnny Cash, John Cash (United States country music singer and songwriter (1932-2003))
    - S: (n) Chevalier, Maurice Chevalier (French actor and cabaret singer (1888-1972))
    - S: (n) Dietrich, Marlene Dietrich, Maria Magdalene von Losch (United States film actress (born in Germany) who made many films with Josef von Sternberg and later was a successful cabaret star (1901-1992))
    - S: (n) Dylan, Bob Dylan (United States songwriter noted for his protest songs (born in 1941))
    - S: (n) Fitzgerald, Ella Fitzgerald (United States scat singer (1917-1996))
    - S: (n) Garland, Judy Garland (United States singer and film actress (1922-1969))
    - S: (n) Horne, Lena Horne, Lena Calhoun Horne (United States singer and actress (born in 1917))
    - S: (n) Iglesias, Julio Iglesias (Spanish singer noted for his ballads and love songs (born in 1943))
    - S: (n) Jackson, Mahalia Jackson (United States singer who did much to popularize gospel music (1911-1972))
    - S: (n) Jackson, Michael Jackson, Michael Joe Jackson (United States singer who began singing with his four brothers and later became a highly successful star during the 1980s (born in 1958))

only 32 singers !?
4 guitarists
5 scientists
0 enterprises
2 entrepreneurs

WordNet classes lack instances

Wikipedia

Copyright Ellis Horowitz 2011-2022

Wikipedia: Tranformation from Database to KnowledgeBase

Wikipedia's Original Mission Statement

"Imagine a world in which every person on the planet shares in the sum of all human knowledge. That is what we're doing.

(for free, in the language of their choice)

Wikipedia's Scale

- **As of April, 2019 Wikipedia's database when dumped takes up 100GBs compressed, 10TBs uncompressed**
- **Total wiki pages:** 51,000,000+
- **Total English articles:** 6.1 million
- **Unique visitors per month:** 500 million
- **Monthly mobile page views:** 3.7 billion

Jimmy Wales, Founder

By Mikael Häggström ,

Wikipedia's Five Pillars (5P)

1. **Encyclopaedia**
   - Notable topics
   - No original research (NOR)
2. **Neutral point of view (NPOV)**
   - Verifiability (referencing)
3. **Free content**
   - Anyone can edit
   - No copyright infringements
4. **Be civil**
5. **No firm rules**

Polypompholyx Talk Sandbox Preferences Watchlist Contributions Log out

Project page Discussion Read Edit View history Search

WIKIPEDIA The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random page
Donate

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Toolbox

Print/export

Languages
Alemannisch
Ænglisc
العربية
অসমীয়া
Azərbaycanca
বাংলা
Беларуская
Беларуская (тарашкевіца)
Boarisch
Bosanski
Brezhoneg
Български
Català
Čeština
Cymraeg
Dansk
Deutsch
ھٕرٛوٕرٛتھ
Ελληνικά
Español
Esperanto
Euskara
فارسی
Français
Furlan
Galego
گیلکی
한국어

Wikipedia:Five pillars

From Wikipedia, the free encyclopedia

The fundamental principles by which Wikipedia operates can be summarized in the form of five "pillars":

Shortcuts: WP:FIVE WP:5P WP:FIVEPILLARS

Wikipedia is an encyclopedia.

It incorporates elements of general and specialized encyclopedias, almanacs, and gazetteers. Wikipedia is not a soapbox, an advertising platform, a vanity press, an experiment in anarchy or democracy, an indiscriminate collection of information, or a web directory. It is not a dictionary, a newspaper, or a collection of source documents; that kind of content should be contributed instead to the Wikimedia sister projects.

Wikipedia is written from a neutral point of view.

We strive for articles that document and explain the major points of view in a balanced and impartial manner. We avoid advocacy and we characterize information and issues rather than debate them. In some areas there may be just one well-recognized point of view; in other areas we describe multiple points of view, presenting each accurately and in context, and not presenting any point of view as "the truth" or "the best view". All articles must strive for verifiable accuracy: unreferenced material may be removed, so please provide references. Editors' personal experiences, interpretations, or opinions do not belong here. That means citing verifiable, authoritative sources, especially on controversial topics and when the subject is a living person.

Wikipedia is free content that anyone can edit, use, modify, and distribute.

Respect copyright laws, and do not plagiarize sources. Non-free content is allowed under fair use, but strive to find free alternatives to any media or content that you wish to add to Wikipedia. Since all your contributions are freely licensed to the public, no editor owns any article; all of your contributions can and will be mercilessly edited and redistributed.

Editors should interact with each other in a respectful and civil manner.

Respect and be polite to your fellow Wikipedians, even when you disagree. Apply Wikipedia etiquette, and avoid personal attacks. Find consensus, avoid edit wars, and remember that there are 4,143,499 articles on the English Wikipedia to work on and discuss. Act in good faith, and never disrupt Wikipedia to illustrate a point. Be open and welcoming, and assume good faith on the part of others. When conflict arises, discuss details on the talk page, and follow dispute resolution.

Wikipedia does not have firm rules.

Rules in Wikipedia are not carved in stone, as their wording and interpretation are likely to change over time. The principles and spirit of Wikipedia's rules matter more than their literal wording, and sometimes improving Wikipedia requires making an exception to a rule. Be bold (but not reckless) in updating articles and do not worry about making mistakes. Prior versions of pages are saved, so any mistakes can be corrected.

v · t · e	Wikipedia principles	[show]
v · t · e	Key Wikipedia policies and guidelines	[show]

Category (±): Wikipedia basic information (–) (±) | (+)

This page was last modified on 15 December 2012, at 03:41.

Text is available under the Creative Commons Attribution/Share-Alike License; additional terms may apply. See Terms of use for details.

Privacy policy About Wikipedia Disclaimers Mobile view

Wikipedia Statistics

- Among top 10 most visited websites
- 70% of traffic is from search engines
- Cited in hundreds of U.S. court rulings

https://stats.wikimedia.org/EN/Sitemap.htm
and
https://stats.wikimedia.org/#/all-projects

Copyright Ellis Horowitz, 2011-2

Wikipedia is a Rich Source of Instances

Wikipedia founders

Jimmy Wales

Larry Sanger

Steve Jobs

From Wikipedia, the free encyclopedia

For the biography, see Steve Jobs (biography).

Steven Paul Jobs (/ˈdʒɒbz/; February 24, 1955 – October 5, 2011)[4][5] was an American businessman and inventor widely recognized as a charismatic pioneer of the personal computer revolution.[6][7] He was co-founder, chairman, and chief executive officer of Apple Inc. Jobs also co-founded and served as chief executive of Pixar Animation Studios; he became a member of the board of directors of The Walt Disney Company in 2006, following the acquisition of Pixar by Disney.

In the late 1970s, Apple co-founder Steve Wozniak engineered one of the first commercially successful lines of personal computers, the Apple II series. Jobs directed its aesthetic design and marketing along with A.C. "Mike" Markkula, Jr. and others. In the early 1980s, Jobs was among the first to see the commercial potential of Xerox PARC's mouse-driven graphical user interface, which led to the creation of the Apple Lisa (engineered by Ken Rothmuller and John Couch) and, one year later, creation of Apple employee Jef Raskin's Macintosh.

After losing a power struggle with the board of directors in 1985, Jobs left Apple and founded NeXT, a computer platform development company specializing in the higher-education and business markets. NeXT was eventually acquired by Apple in 1996, which brought Jobs back to the company he co-founded, and provided Apple with the NeXTSTEP codebase, from which the Mac OS X was developed."[8] Jobs was named Apple advisor in 1996, interim CEO in 1997, and CEO from 2000 until his resignation. He oversaw the development of the iMac, iTunes, iPod, iPhone, and iPad and the company's Apple Retail Stores.[9] In 1986, he acquired the computer graphics division of Lucasfilm Ltd, which was spun off as Pixar Animation Studios.[10] He was credited in *Toy Story* (1995) as an executive producer. He remained CEO and majority shareholder at 50.1 percent until its acquisition by The Walt Disney Company in 2006,[11] making Jobs Disney's largest individual shareholder at seven percent and a member of Disney's Board of Directors.[12][13]

In 2003, Jobs was diagnosed with a pancreas neuroendocrine tumor. Though it was initially treated, he reported a hormone imbalance, underwent a liver transplant in 2009, and appeared progressively thinner as his health declined.[14] On medical leave for most of 2011, Jobs resigned as Apple CEO in August that year and was elected Chairman of the Board. On October 5, 2011, Jobs died of respiratory arrest related to his metastatic tumor. He

Steve Jobs

Jobs holding a white iPhone 4 at Worldwide Developers Conference 2010

Born	Steven Paul Jobs February 24, 1955[1][2] San Francisco, California, U.S.[1][2]
Died	October 5, 2011 (aged 56)[2] Palo Alto, California, U.S.
Nationality	American
Alma mater	Reed College (dropped out)

Wikipedia's Categories Also Contain Classes

Categories: Steve Jobs | 1955 births | 2011 deaths | American adoptees | American billionaires | American chief executives | American computer businesspeople | American industrial designers | American inventors | American people of German descent | American people of Swiss descent | American people of Syrian descent | American technology company founders | American Zen Buddhists | Apple Inc. | Apple Inc. employees | Businesspeople from California | Businesspeople in software | Cancer deaths in California | Computer designers | Computer pioneers | Deaths from pancreatic cancer | Disney people | Internet pioneers | National Medal of Technology recipients | NeXT | Organ transplant recipients | People from the San Francisco Bay Area | Pescetarians | Reed College alumni

But categories do not form a taxonomic hierarchy, i.e. there is no ISA hierarchy

An isa hierarchy only specifies that a set of objects is the subclasses of another object, but nothing more

Structure of a Wikipedia Page

- Types of links
  - *Article links*
    - links from one article to another of the same language;
  - *Category links*
    - links from an article to special "Category" pages;
  - *Interlingual links*
    - links from an article to a presumably equivalent, article in another language;
- Types of special pages
  - *Redirect pages*
    - short pages which often provide equivalent names for an entity
  - *Disambiguation pages*
    - a page with little content that links to multiple similarly named articles.
- Infoboxes, templates, list pages, wikipedia commons, ...

Combining Wikipedia Named Entities to WordNet Synsets

See "*Mapping WordNet synsets to Wikipedia*" Fernando & Stevenson, and "Mapping WordNet Instances to Wikipedia", McCrae

Wikimedia Foundation

- Wikimedia Foundation, Inc. (WMF) is an American non-profit and charitable organization headquartered in San Francisco.
- It owns the internet domain names and hosts Wikipedia
- The foundation was founded in 2003 by Jimmy Wales as a way to fund Wikipedia and its sister projects through non-profit means
- As of 2015, the foundation employs over 280 people,

with annual revenues in excess of $75 million

- Related projects to Wikipedia:
  - **Commons** for multimedia,
  - **Wiktionary** as free dictionary, and
  - **Wikidata** for structured data.

WikiData

- WikiData is an effort to convert the Wikipedia data into a knowledgebase
- WikiData aims to create a free RDF-like KB about the world that can be read/edited by humans & machines
- Wikidata clients use the repository, e.g. to populate Web pages or Wikipedia infoboxes
- WikiData increases the quality and lowers the maintenance costs of Wikipedia and related projects

Wikidata Multilingual Coverage

- The challenge: Wikipedia has many named entities that occur in numerous articles
  - E.g Ex-President Obama or President Trump are mentioned in over 100 articles
  - If one of them dies, this must be reflected on many of those pages
- Solution: Let the entry for Obama/Trump be centered in Wikidata and let all references to Obama/Trump point to the Wikidata entry
- Another aspect of Wikidata is their multilingual coverage
  - Popular entities are present in many languages (up to 180); and even in one Wikipedia page there may be many languages
  - E.g. Lucas Cranach (German Renaissance Painter) in Wikidata is referenced in 57 language tags, representing 44 languages and 13 language variants

Wikipedia Page and WikiData Page for Douglas Adams

Article Talk Read Edit View history Search Wikipedia

Douglas Adams

From Wikipedia, the free encyclopedia
(Redirected from Douglas adams)

For other people named Douglas Adams, see Douglas Adams (disambiguation).

Douglas Noel Adams (11 March 1952 – 11 May 2001) was an English author, scriptwriter, essayist, humorist, satirist and dramatist.

Adams was author of *The Hitchhiker's Guide to the Galaxy*, which originated in 1978 as a BBC radio comedy before developing into a "trilogy" of five books that sold more than 15 million copies in his lifetime and generated a television series, several stage plays, comics, a computer game, and in 2005 a feature film. Adams's contribution to UK radio is commemorated in The Radio Academy's Hall of Fame.[1]

Adams also wrote *Dirk Gently's Holistic Detective Agency* (1987) and *The Long Dark Tea-Time of the Soul* (1988), and co-wrote *The Meaning of Liff* (1983), *The Deeper Meaning of Liff* (1990), *Last Chance to See* (1990), and three stories for the television series *Doctor Who*; he also served as script editor for the show's seventeenth season in 1979. A posthumous collection of his works, including an unfinished novel, was published as *The Salmon of Doubt* in 2002.

Adams was an advocate for environmentalism and conservation, a lover of fast cars,[2] technological innovation and the Apple Macintosh, and a self-proclaimed radical atheist.

Contents [hide]

1 Early life
1.1 Education
2 Career
2.1 Writing
2.1.1 *Doctor Who*
2.1.2 *The Hitchhiker's Guide to the Galaxy*
2.1.3 *Dirk Gently* series
2.2 Music
2.2.1 Pink Floyd
2.3 Computer games and projects

Douglas Adams

Born	Douglas Noel Adams 11 March 1952 Cambridge, England
Died	11 May 2001 (aged 49) Montecito, California, US
Occupation	Writer
Alma mater	St John's College, Cambridge
Genre	Science fiction, comedy, satire
Notable work	*The Hitchhiker's Guide to the Galaxy*
Signature	
Website	douglasadams.com

His Wikipedia page

His WikiData page

Statements in Wikidata consist of key-value pairs, which match a **property** (such as "author", or "publication date") with one or more **values** (such as “Sir Arthur Conan Doyle” or "1902").

Copyright Ellis Horowitz 2011-2022

Google's Knowledge Graph

Copyright Ellis Horowitz 2011-2022

Google Knowledge Graph

- Introduced in 2012 with the Hummingbird update, see https://searchengineland.com/google-hummingbird-172816
- Powered in part by Freebase
- KnowledgeGraph was accused of taking away traffic from Wikipedia
- Knowledge panels are information boxes for entities (person, place, organization, event, etc)
- A common source of info is Wikipedia, LinkedIn, Crunchbase, Reuters, Bloomberg

https://www.google.com/intl/es419/insidesearch/features/search/knowledge.html

Google's slogan for the knowledge graph: "things, not strings"

https://www.youtube.com/watch?v=mmQl6VGvX-c

Copyright Ellis Horowitz, 2011-2022

Knowledge Graph Enhances Google Search in 3 main ways (1):

mausoleum

musician

1. To improve the variety of search results, Google uses the knowledge graph to locate alternate interpretations of query terms,

Here it offers two of them with the same name e.g.
"taj mahal" - the mausoleum or musician

Copyright Ellis Horowitz, 2011-2022

Knowledge Graph Enhances Google Search in 3 main ways (2):

2. To provide deeper and broader results, typically in an info box
e.g. person entities include relations such as age, birthplace, marital status, children, education, etc.,
here is a sample result for Matt Groening

- creator of The Simpsons
- **Go Deeper**
  - his photo
  - when he was born
  - his spouse
  - his parents
  - why he is famous
- **Go Broader**
  - other people related to Groening

Copyright Ellis Horowitz 2011-2022

Knowledge Graph Enhances Google Search in 3 main ways: (3)

3. To provide the best summary
the knowledge graph exploits the relationships among the entities
e.g. the query "Tesla"

The knowledge graph allows Google to summarize relevant content around that topic, including key facts you're likely to need for that particular thing. E.g.
Tesla Motors, Inc. is an American automotive and energy storage company that designs, manufactures, and sells luxury electric cars, electric vehicle powertrain components, and battery produc

Copyright Ellis Horowitz 2011-2022

KnowledgeGraph Internal Representation Example

Google Search Combines Document Index with Knowledge Graph

When Google recognizes in the search query that it is about an entity recorded in the KnowledgeGraph it focuses its search around the entity

Query Processing

Copyright Ellis Horowitz 2011-2022

3 Parts to Today's Lecture

1. **Restructuring the inverted index to speed up processing**
   - See Chapter 7 of our textbook
2. **Reverse engineering Google's query processing algorithm**
3. **A close up look at Google's internal architecture**

Copyright Ellis Horowitz, 2011-2022

Speeding Up Indexed Retrieval

- **User has a task and formulates it as a query**
- **The search engine's task is to**
  1. **Minimally return documents that contain the query terms**
     - Use inverted index and cosine similarity to identify matching documents
     - Try to identify the K top scoring documents and return those
  2. **Determine what the user is actually trying to accomplish, even though the query may be (at best) vaguely stated**
     - Use knowledge graph, user location, profile, etc to create the most likely responses
- **The following slides contain heuristics that can be applied to speed up step 1 of the process**

Copyright Ellis Horowitz, 2011-2022

Strategy 1: Consider Only Query Terms with High-idf Scores

- **For a query such as *catcher in the rye***
- **Only accumulate (cosine) scores for *catcher* and *rye***
- **Intuition: *in* and *the* contribute little to the scores and so <u>don't alter rank-ordering much</u>**
- **Benefit:**
  - Postings of low-idf terms have many docs, but these many docs will eventually get eliminated from the set of contenders, so it is best to avoid them initially

Copyright Ellis Horowitz, 2011-2022

Strategy 2: Consider Only Docs Containing Several Query Terms

- In theory, any doc with at least one query term is a candidate for the output list
- However, for multi-term queries, only compute cosine scores for docs containing several of the query terms
  - Say, at least 3 out of 4
  - This imposes a "soft conjunction" on queries seen on web search engines
- Easy to implement in postings traversal

Copyright Ellis Horowitz, 2011-2022

Strategy 3: Introduce Champion Lists Heuristic

- **Pre-compute for each dictionary term *t*, the *r* docs of highest weight (tf-idf) in *t*'s postings**
  - Call this the <u>champion list</u> for *t*
  - (aka <u>fancy list</u> or <u>top docs</u> for *t*)
- **Note that *r* has to be chosen at index build time**
  - Thus, it's possible that $r < K$ *(number of docs to return)*
  - The value of *r* can vary for different terms
- **At query time, only compute scores for docs in the champion list of some query term**
  - champion lists that include the terms of the query
  - Pick the *K* top-scoring docs from among these

Copyright Ellis Horowitz, 2011-2022

Static Quality Scores Heuristic

- We want top-ranking documents to be both *relevant* and *authoritative*
- *Relevance* is being modeled by cosine scores
- *Authority* is typically a query-independent property of a document
- Examples of authority signals
  - Wikipedia among websites
  - Articles in curated newspapers
  - A paper/webpage with many citations, or equivalently
  - A web page with high PageRank

Copyright Ellis Horowitz, 2011-2022

Strategy 4: Introduce an Authority Measure

- **Assign to each document *d a query-independent* quality score in [0,1]**
- **Denote this by *g(d), g stands for goodness***
- ***Authority measures might include:***
  - Documents with a high number of citations scaled into the range of [0,1]
  - Documents with high PageRank, also can be scaled in [0,1]
  - Heavily curated content, e.g. Wikipedia, NY Times
  - Documents with many favorable reviews

Copyright Ellis Horowitz, 2011-2022

Combine Relevance and Authority

- **Consider a simple total score combining cosine relevance and authority**
- **net-score(q,d) = $g(d)$ + cosine(q,d)**
  - **For query q and document d its net-score is a combination of authority and relevance**
  - **We could use some other linear combination, assigning different weights to the two factors;**
  - **In processing a query the algorithm now seeks the top K docs by <u>net-score</u>**

Copyright Ellis Horowitz, 2011-2022

Strategy 5: Reorganize the Inverted List

- So far we assumed that all documents were ordered by docID, even those on the champion lists
- Instead order all postings by *g(d)* the authority measure
- This does not change the earlier computing times for merging
- The most authoritative documents will appear early in the postings list
- Thus, can concurrently traverse query terms' postings for
  - Postings intersection
  - Cosine score computation

Copyright Ellis Horowitz, 2011-2022

Computing Net Score

- Combine champion lists with *g(d)*-ordering
- Maintain for each term a champion list of the *r* docs with highest *g(d)* + tf-idf$_{td}$
- Seek top-*K* results from only the docs in these champion lists

authority score relevance cosine score

- This is equivalent to

$$\text{net-score}(q,d) = g(d) + \frac{\vec{V}(q)\cdot\vec{V}(d)}{|\vec{V}(q)||\vec{V}(d)|}.$$

Copyright Ellis Horowitz, 2011-2022

Strategy 6: High and Low Lists Heuristic

- **For each term, maintain two postings lists called *high* and *low***
  - Think of *high* as the champion list
- **When traversing postings on a query, only traverse *high* lists first**
  - If we get more than *K* docs, select the top *K* and stop
  - Else proceed to get docs from the *low* lists
- **Can be used even for simple cosine scores, without global quality *g(d)***
- **This assumes we have a means for segmenting index into two <u>tiers</u>**

Copyright Ellis Horowitz, 2011-2022

Example of a Tiered Inverted Index; A Generalization of Champion Lists

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

4 documents with term frequencies

term	df_t	idf_t
car	18,165	1.65
auto	6723	2.08
insurance	19,241	1.62
best	25,235	1.5

Inverse document frequencies

Static quality-ordered index; Assume doc1, doc2, doc3 have quality scores *g(1)=0.25, g(2)=0.5, g(3)=1*

Tiered index, threshold of 20 for tier 1, 10 for tier 2 if tier 1 doesn't provide enough results, try tier 2, etc

Copyright Ellis Horowitz, 2011-2022

Recap: How to Compute Cosine Score

Dot product

Unit vectors

$$\cos(\vec{q},\vec{d}) = \frac{\vec{q} \bullet \vec{d}}{|\vec{q}||\vec{d}|} = \frac{\vec{q}}{|\vec{q}|} \bullet \frac{\vec{d}}{|\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

$cos(\vec{q},\vec{d})$ is the cosine similarity of $\vec{q}$ and $\vec{d}$ … or, equivalently, the cosine of the angle between $\vec{q}$ and $\vec{d}$.

The algorithm for computing cosine scores can be found in Figure 6.14 of our textbook

Copyright Ellis Horowitz, 2011-2022

Part 2: Google's Query Processing Algorithm

- **Now let's switch gears and look at the problem of reverse engineering Google's query processing (ranking) algorithm**
- **There are two main companies trying to do this:**
  1. ***Searchmetrics* which tracks the results from thousands of keywords while analyzing the content on each page producing a ranking that determines what content elements are most important in Google's ranking algorithm**
  2. ***Moz.com* which also tracks the results from thousands of keywords and then measures how the website rankings changed whenever Google performs an update to its ranking algorithm**

Copyright Ellis Horowitz, 2011-2022

Searchmetrics Tracks Ranking Factors Valued by Google's Algorithm

Reverse engineering the Google ranking algorithm

Download at:
http://csci572.com/papers/Searchmetrics.pdf

Why search is important commercially

Copyright Ellis Horowitz 2011-2022

General Ranking Factors

Here are the major categories and some factors

Content

Overall relevance
Word count

User Signals

CTR
Bounce rate

Technical

HTTPS
Presence of H1/H2

User Experience

Font size
number of internal/external links

Social Signals

Pinterest
Facebook/Tweet

Backlinks

number of No-follow

Copyright Ell

Note: What Are Nofollow Links

- The nofollow tag tells search engines to ignore that link
- Google introduced the rel="nofollow" option in 2005 for bloggers that were struggling with people using comment spam to try and build links in the hope of ranking for specific keywords
- Examples
- `<a href="https://example.com" rel="nofollow">Anchor Text</a>`
- In September 2019, Google announced two new link attributes that offered webmasters additional ways to help Google identify the nature of different links
  - `rel = “sponsored”`
  - identify links on your site that were created as part of advertisements
  - `rel = “ugc”`
  - Google recommends marking user-generated content (UGC) links, such as comments and forum posts, as UGC
- **Other relevant attributes**
  - `<meta name="robots" content="nofollow" />`
  - `<meta name="robots" content="noindex, follow">;`

Copyright Ellis Horowitz, 2011-2022

Content Factors

The analysis shows that the content relevance, decreases as the position in the search results drops.

The highest content relevance scores were found among the results for positions 3 to 6.

Thereafter, the landing pages on subsequent positions show lower relevance scores

Copyright Ellis Horowitz 2011-2022

Word Count

- The word count of a landing page ranked among the top positions
- Pages rank well under the condition that the content is not simply long, but also relevant,

Copyright Ellis Horowitz 2011-2022

Click-Through Rate

The Click-Through Rate measures the average percentage of users who click on the result at each position on the SERP*.

Keywords in position 1 have an average CTR of 44%, the rate dropping to 30% for position 3.

The click rate for landing pages at the top of the second results page is higher than for results at the bottom of page 1.

*SERP: Search Engine Results Page

Copyright Ellis Horowitz 2011-2022

Bounce Rate

The Bounce Rate measures the percentage of users who only click on the URL from Google's search results, without visiting any other URLs at that domain, and then return back to the SERP*.

These are single-page sessions where the user leaves the site without interacting with the page.

*SERP: Search Engine Result Page

Copyright Ellis Horowitz 2011-2022

Technical Factors

- Page encryption using HTTPS is rising.
- In 2017 only 12% of pages relied on data transfer via HTTP. Today, this has more than tripled, with over a third of websites encrypting the data traffic on their pages
- In 2017 Google announced that pages that have not switched to HTTPS would be marked as "unsafe" in its Chrome browser.

Copyright Ellis Horowitz 2011-2022

Social Signals

- The correlation between social signals and ranking position is extremely high
- Facebook remains the social network with by far the highest level of user interactions.
- Facebook, compared with the other social networks, shows relatively high signals across the first search results page

Copyright Ellis Horowitz 2011-2022

Mobile Friendliness

All top 100 websites have a mobile-friendly version; they use either a mobile sub-domain or responsive design;

Separate mobile websites are diminishing in popularity, but e.g. try Sephora.com on desktop and mobile. Last time I looked they were still using m.sephora.com

Over a fifth of websites outside of the top 100 offer no mobile-friendly solution

Mobile-friendly websites

The following graphics show the frequency of websites with mobile-friendly solutions amongst the top 100 domains by SEO visibility.

Top 100 Domains Google US

That's right. All 100 of the top 100 have a mobile-friendly solution. These include the use of a mobile sub-domain, dynamic serving, responsive design and/or mobile apps.

Mobile-friendliness: Sample of smaller domains

Over a fifth of websites outside the top 100, based on a sample of smaller domains, offer no mobile-friendly solution to smartphone users. The upcoming shift to a mobile-first index will have a negative impact on such websites, should they fail to react and implement mobile-friendly solutions.

Copyright Ellis Horowitz 2011-2022

Finally Correlation is Not Necessarily Causation

- **See the full Searchmetrics report at**
  - http://csci572.com/papers/Searchmetrics.pdf
- **WARNING: SEO studies always make it clear that their findings may not actually define the way the Google search result algorithm actually works**
  - Correlations are not synonymous with causal relationships
- **There are many examples of illusory correlations which are referred to as "logical fallacy"; here are two examples**
  1. the co-appearance of phenomena like the number of storks and the higher birthrate in certain areas,
  2. the relationship between sales of ice-cream and increased incidence of sunburn in the summer.
- **These examples show a (illusory) correlation, not a causal relationship.**
- **For many more funny ones go to: http://tylervigen.com/spurious-correlations**

Copyright Ellis Horowitz, 2011-2022

Another Way to Reverse Engineer Google's Query Processing Algorithm

- **Moz.com Monitors Google's Ranking Algorithm watching how search results are affected**
- **Google has changed its ranking algorithm many times over the years**
- **This algorithm is especially important to advertisers**
- **Moz.com is an SEO company that sells its software as a service (SaaS); capabilities offered include:**
  - Keyword research
  - Improving your ranking
  - Comparing your site with the competition
- **As part of its service MOZ offers two mechanisms**
  - ***MozRank* scores your web page based upon external criteria**
  - ***MozCast* keeps track of whenever Google changes its ranking algorithm, see https://moz.com/google-algorithm-change**

Copyright Ellis Horowitz, 2011-2022

MozRank

- **MozRank** is a logarithmically scaled 10-point measurement of website linking authority or popularity of a given web page (https://moz.com/help/link-explorer)
  - It could be viewed as analogous to PageRank
  - See 4 minute video on the page
- MozRank is based on a score called the domain authority, a number between 1 and 100

Criteria include:

- Number of links to your site
- Quality of sites you link to
- Number of trusted sites linked to
- Quality of your content
- Social signals referencing your site

Copyright Ellis Horowitz, 2011-2022

Confirmed Google Updates to Their Ranking Algorithm (Part 1)

- https://moz.com/google-algorithm-change
- Select any date to view the changes

- 2003 Feb, Boston, index refresh
- 2003, July, Fritz, switch to incremental index updates
- 2003, Nov, Florida, block keyword stuffing
- 2004, Feb, Brandy, index expansion, use of LSI
- 2005, Jan, Nofollow, introduce nofollow attribute
- 2005, May, Bourbon, improved treating of duplicate content
- 2005, June, Personalized search, keeping user search history
- 2005, June, XML sitemaps
- 2005, Oct, Jagger, eliminating low quality links
- 2005, Oct, Google local maps,
- 2005, Dec BigDaddy, handling URL canonicalization redirects

Copyright Ellis Horowitz, 2011-2022

Confirmed Google Updates to their Ranking Algorithm (Part 3)

- 2018, Mar, Mobile-First Index Roll-out
- 2018, Apr, Unnamed Core Update, no information given
- 2018, May, Snippet Length drop, rolled back snippet length to 150-160 characters
- 2018, Jun, Video Carousels, new display format
- 2018, Jul, Mobile speed update, page speed is a ranking factor
- 2018, Jul, chrome security warning on non-HTTPS sites
- 2018, Aug, Medic Core update, affecting health and wellness sites

- 2019, Mar, core update
- 2019, Apr, May, Jun, Deindexing Bug identified and fixed
- 2019, Jun, Site Diversity Update, reducing multiple occurrences of same site
- 2019, Sep, core Update, no information provided
- 2019, Oct, BERT update, BERT natural language processing model
- 2019, Dec, International rollout of BERT

- 2020, Jan, Core update
- 2020, Jan Featured Snippet De-duping, URLs in featured snippets no longer appearing as organic results
- 2020, May Core update
- 2020, Aug, Google Glitch, broken and then quickly fixed
- 2020, Sep, Oct, bug fixes

Copyright Ellis Horowitz, 2011-2022

Moz.com Tracks Google Algorithm Updates

- MozCast is a statistical technique designed to highlight the affects of Google modifying their ranking algorithm
- Every 24 hours, Moz tracks a hand-picked set of 1,000 keywords and grab the top 10 Google organic results. Keywords were deliberately chosen to avoid obvious local intent, are distributed evenly across 5 "bins" by query volume, and are tracked at roughly the same time every day from the same location
- Each day, they take the current top 10 and compare it to the previous day's top 10 (for any given keyword) and calculate a rate of change or "delta".
- This is done across all 1,000 keywords; they express the result as a “temperature in Farenheit; an average day is about 70° F.

Copyright Ellis Horowitz, 2011-2022

The Google Architecture

See Google's Website on how search works at

http://www.google.com/insidesearch/howsearchworks/thestory/

Much of these notes are based upon Keith Erekson's CSE497 and C. Lee Giles from Penn State IST 441 and Jeff Dean's Slides on Google

Copyright Ellis Horowitz 2011-2022

How Google Search Has Changed Over the Years

2001, adds “did you mean”
2002, handles synonyms
2004, added news & stock quotes
2005, added Autocomplete
2006, added video, weather, flights
2007, added movie times & patents
2008, Google search mobile app
2009, voice search
2010, Google Instant
2011, added image search
2012, added knowledge graph
2013, use of carousels for display

Copyright Ellis Horowitz 2011-2022

Information Retrieval Challenges for Google from 1999 to 2014

- According to Jeff Dean, Google fellow

how things have changed:

- #doc: ~70 million to many billion ~100X
- # queries processed/day ~1000X
- size of the document index ~3X
- update latency: months to minutes ~10000X
- average query latency: <1s to <0.15s ~6X
- more machines ~1000X

Copyright Ellis Horowitz, 2011-2022

Google Serving System circa 1999

A database **shard** is a horizontal partition of data in a database or search engine. Each individual partition is referred to as a **shard**. Each **shard** is held on a separate database server instance, to spread the load.

Copyright Ellis Horowitz 2011-2022

Original Google Architecture Diagram

Logical Entities

- URL Server
- Crawler – across multiple machines
- Store Server
- Repository – all web pages
- Indexer – parses documents
- URL Resolver – converts relative URLs
- Barrels – contain words in documents
- Sorter – takes barrels sorted by document and re-sorts by word
- Lexicon – word/phrase index

Figure 1. High Level Google Architecture

Diagram from "*The Anatomy of a Large-Scale Hypertextual Web Search Engine*" http://infolab.stanford.edu/~backrub/google.html

Google's Early Query Processing Basic Steps

1. Parse the query
2. Convert words into wordIDs using the lexicon
3. Select the barrels that contain documents which match the wordIDs
4. Scan through the document list until there is a document that matches all of the search terms
5. Compute the rank of that document for the query (using PageRank as one component)
6. Repeat step 4 until no documents are found and we've examined all of the barrels
7. Sort the set of returned documents that have been matched by document rank and return the top k.

Modern Query Processing Methodology

- Google (and others) have now moved query processing far beyond keyword matching
- In ***semantic*** information retrieval systems, entities play a central role in several tasks.

1. Understanding the search query (Search Query Processing)
2. Relevance determination at document level (scoring)
3. Compilation of search results and SERP* features

- ***Important steps***

1. Identification of the thematic ontology in which the search query is located
   - If the thematic context is clear, Google can select a content corpus of text documents, videos, images … as potentially suitable search results.
2. Identification of entities and their meaning in the search term (named entity recognition)
3. Understanding the semantic meaning of a search query.
4. Identification of the search intent
5. Semantic annotation of the search query
6. Refinement of the search term

- *Search engine results page

Copyright Ellis Horowitz, 2011-2022

Term-Based versus Entity-Based Meaning

An entity based search recognizes the different context based on the different arrangement; "Stoplight red" is an entity, not to be confused as separate terms "red" and "stoplight"

Copyright Ellis Horowitz 2011-2022

Google's Query Processing Elements

1. Rankbrain/MUM/Bert are used to identify entities
2. Knowledge Graph: an ontology
3. Named Entity Recognition
4. Document Index: inverted index of terms
5. Relevance Scoring
6. Duplicate removal
7. Personalization

© Olaf Kopp, Aufgesang GmbH

Using the KnowledgeGraph to Identify Entities

- The biggest challenge for Google with regard to semantic search is identifying and extracting entities, their attributes and other information from data sources such as websites.
- The information is mostly not structured and not error-free.
- The current Knowledge Graph is largely based on the structured content from Wikidata and the semistructured data from Wikipedia or Wikimedia.

Copyright Ellis Horowitz, 2011-2022

Entity Recognition in the Knowledge Graph: Two Examples

Query: ceo vw
No where is the person's name mentioned
Entities are: "vw" and "boss"

Query: founder adidas
Entities: "Adidas", "founder"

Copyright Ellis Horowitz 2011-2022

Google Patent Describing Semantic Clustering

- Terms are assigned to entities and to a thematic context
  - E.g. the term "mercedes" to ID3279 and its context is "Auto"
- Patent title: *Improving topic clustering on search queries with word co-occurrence and bipartite graph co-clustering* is based on this paper
  - https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45569.pdf
  - The notion of "lift" scores is introduced using this formula

$$lift(\mathrm{w}_i; \mathbf{a}) = \frac{P(\mathrm{w}_i \mid \mathbf{a})}{P(\mathrm{w}_i)},$$

- w_i is in the formula for all terms that are closely related to the root of the word, such as misspellings, plural, singular or synonyms.
- ***"a"*** can be any user interaction such as searching for a specific search term or visiting a specific page.
- For example, if the lift score is 5, the probability that "Wi" will be searched is 5 times higher than that of "Wi" being searched in general.
-

Copyright Ellis Horowitz, 2011-2022

RankBrain
An Entity-Based Processor

- RankBrain is a deep learning-based algorithm that is used after the selection of an initial subset of search results
- Introduced in 2015 as part of their Hummingbird algorithm update
  - RankBrain ***maps keywords into entities*** which are then looked for in the Knowledge Graph;
  - Words surrounding the entity are considered the context of the query
  - Google claims that RankBrain is the third most important factor in their ranking algorithm (links/words being numbers 1 and 2)

RankBrainExample

Above are two queries using similar phrases "level of the food chain"; In the first query the word "consumer" can be misinterpreted as someone who eats rather than as a predator;
RankBrain will associate the results of the 2nd query to produce more favorable results for the first query

Copyright Ellis Horowitz 2011-2022

RankBrain in Its Simplest Form

1. Google receives a query for something it's never seen before (like a new movie title or a phrase connecting two topics, like "which country has the best cars")
2. Google assigns the entity a unique identifier, like 9202a8c04000641f8000000000006567.
3. Google determines the entity's relatedness to other entities, then assigns it a value.
4. Google determines the entity's notability, then assigns it a value.
5. Google determines the entity's contribution, then assigns it a value.
6. Google evaluates any awards the entity received, then assigns a value.
7. Each value is weighted according to the entity's query type. For example, in the case of the best car example, Google may prioritize relatedness and awards for particular brands, and return the result as a carousel of options rather than a single webpage.

- The strength of RankBrain is it's **ability to** handle novel queries. In addition RankBrain is a framework for continual learning of entities.

Copyright Ellis Horowitz, 2011-2022

What is RankBrain observing exactly?
It's paying very close attention to **how you interact with the search results**. Specifically, it's looking at:

- Organic Click-Through-Rate
- Dwell Time
  - Dwell Time is the amount of time that a Google
  - searcher spends on a page from the search results
  - before returning back to the SERPs.
- Bounce Rate

  Bounce Rate is defined as the percentage of visitors that leave a webpage without taking an action, such as clicking on a link, filling out a form, or making a purchase.
- Pogo-sticking

  Pogo sticking is when a search engine users visits several different search results in order to find a result that satisfies their search query

- These are known as user experience signals (UX signals).

Copyright Ellis Horowitz 2011-2022

RankBrain Patent

Using concepts as contexts for query term substitutions

- a method includes receiving a query that includes at least three sequential query terms; determining that the sequential query terms represent a concept; and in response to determining that the sequential query terms represent a concept, collecting query term substitution data for one or more query terms that occur in queries that include the concept.

Copyright Ellis Horowitz, 2011-2022

RankBrain Example

- A search query like "Action movie starring Tom Cruise"
- First each term is identified: "Action", "Movie", "Tom" and "Cruise".
  - The term Action can be an entity of the entity type Movie, Genre or Series.
  - The one that is selected is based upon its frequency or popularity of an entity score.
  - That produces the table to the right
  - Depending upon which type of search engine is used (e.g. image search) appropriate media are returned that match "Action Film" and "Tom Cruise"

ENTITY RESULTS

Entity Name	Entity Type	Entity Score
action	SERIES	5.750
action	GENRE	1.036
action	MOVIE	0.345
movies	APP	10.000
movies	CORPUS_TYPE_MOVIE	5.000
movies	CHANNEL	1.000
movies	SERIES	0.000
tom cruise	ACTOR	5.750
tom	WEB_SITE	1.036
tom	SERIES	0.345
tom	MOVIE	10.000
the cruise	MOVIE	5.000
the cruise	SERIES	1.000

Copyright Ellis Horowitz, 2011-2022

RankBrain Offline

- When offline, RankBrain is given
  - batches of past searches and learns by matching search results
  - Newspaper articles and learns to associate items within a news article
- Studies showed how RankBrain better interpreted the relationships between words.
  - RankBrain includes the analysis of stop words as they can affect the meaning of a query, e.g. "flights ***to*** London"
  - RankBrain learns phrases specifically for the country, as well as language, in which a query is made.
  - E.g. a query with the word "boot" in it within the United States, one will get information on footwear. However, if the query comes through the UK, then the information could also be in regards to storage spaces in cars

Copyright Ellis Horowitz, 2011-2022

RankBrain Learns the Concept of Capital Cities

- **Using offline sources RankBrain is able to identify these associations**

The figure illustrates the ability of the model to automatically organize concepts and learn implicitly the relationship between them, as during the training no supervised info was input about what a capital city means

Copyright Ellis Horowitz, 2011-2022

Lucene, Solr and HW#4

Copyright Ellis Horowitz 2011-2022

What is Lucene

- **High performance, scalable, full-text search library**
- **Basically one adds documents to Lucene which creates an inverted index**
  - A document is defined as a collection of fields
  - Lucene includes flexible text analysis including tokenizers and filters
- **Lucene does NOT include crawlers or explicit searching of documents**
  - Searching the index must be implemented elsewhere
- **100% Java, no dependencies**
- **Offered by the Apache Software Foundation**

Copyright Ellis Horowitz, 2011-2022

What is Solr

- A full text search server based on Lucene
- Communicates with Lucene via XML/HTTP, JSON Interfaces
- Flexible data schema to define types and fields
- Output includes highlighting matches
- Configurable advanced caching
- Index Replication
- Extensible Open Architecture, Plugins
- Web Administration Interface
- Written in Java5, deployable as a WAR

Copyright Ellis Horowitz, 2011-2022

Top-Level Overview of Lucene/Solr

Downloading Solr

Lucene
home page

Solr
home page
Download
Solr

You only need to download Solr
Due to security extensions in version 8.11
We recommend you use version 7.7.3

Copyright Ellis Horowitz 2011-2022

Lucene Internals – Positional Inverted Index

ID	Term	Document : position
1	best	2 : 3
2	blue	1: 3, 3 : 2
3	bright	1 : 2, 3 : 5
4	butterfly	1 : 4
5	breeze	1 : 8
6	forget	2 : 5
7	great	2 : 7
8	hangs	1 : 5
9	needs	3 : 8
10	retire	2 : 11
11	search	3 : 10
12	sky	2 : 8, 3 : 3
13	wind	2 : 14

takes documents, parses them into terms and builds a positional inverted index;

Copyright Ellis Horowitz, 2011-2022

Lucene Indexing Pipeline

Analyzer

- **Analyzer : create tokens using a Tokenizer and/or applying Filters (Token Filters)**
  - **Splits words at punctuation characters, removing punctuation. However, a dot that's not followed by whitespace is considered part of a token.**
  - **Splits words at hyphens, unless there's a number in the token, in which case the whole token is interpreted as a product number and is not split.**
  - **Recognizes email addresses and internet hostnames as one token.**

Copyright Ellis Horowitz, 2011-2022

Two Lucene Scoring Concepts

TF – IDF & cosine similarity

Lucene scores using a combination of TF-IDF and vector closeness

1

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

TF - IDF = Term Frequency X Inverse Document Frequency

2

cosine-similarity(query_vector, document_vector) = V(q) * V(d)/|V(q)|*|V(d)|
where V(q)*V(d) is the dot product of the weighted vectors and |V(q)|, |V(d)| are the Euclidean norms of the vectors (square root of the sum of squares)

for details see
https://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html

Copyright Ellis Horowitz, 2011-2022

Solr in Production

Includes support for
-Sharding
-Load balancing
-replication

A **database shard** is a horizontal partition of data in a **database** or search engine.
Each individual partition is referred to as a **shard**. Each **shard** is held on a separate **database** server instance, to spread the load.

Copyright Ellis Horowitz, 2011-2022

High Level Overview

Lucene/Solr

there is a Solr server which accepts queries in the form of http requests and returns results as JSON strings

Copyright Ellis Horowitz, 2011-2022

Solr Request and Result

http://localhost:8983/solr/select?q=video&start=0&rows=2&fl=name,price

```
<response><responseHeader><status>0</status>
  <QTime>1</QTime></responseHeader>
  <result numFound="16173" start="0">
    <doc>
      <str name="name">Apple 60 GB iPod with Video</str>
      <float name="price">399.0</float>
    </doc>
    <doc>
      <str name="name">ASUS Extreme N7800GTX/2DHTV</str>
      <float name="price">479.95</float>
    </doc>
  </result>
</response>
```

- status, always 0 in a successful response
- QTime, the server-side query time in milliseconds
- numFound, the total number of documents matching the query
- start, the offset into the ordered list of results
- field types in <doc> include str, boolean, int, long, float, double, date, lst, arr
  - lst is a named list <lst><int name="foo">33</int><int name="bar">42</int></lst>
  - arr is an array <arr><int>33</int><int>42</int></arr>
- multivalued fields are returned in an <arr> element.

Copyright Ellis Horowitz, 2011-2022

How to Start Solr

Complete installation instructions can be found at

https://solr.apache.org/guide/8_11/installing-solr.html

Once it is installed:

1. Start Solr

java -jar start.jar

2. Index your XML data

java -jar post.jar *.xml

3. Search http://localhost:8983/solr

localhost indicates the Solr server is running locally on port 8983

Copyright Ellis Horowitz, 2011-2022

Querying Data and XML Response

- HTTP GET or POST with parameters are used to specify queries
- E.g. here are 4 sample queries, some with various parameters

price and desc are fields

http://solr/select?q=electronics

http://solr/select?q=electronics&sort=price+desc

http://solr/select?q=electronics&rows=50&start=50

http://solr/select?q=electronics&fl=name+price

limit results to fields: name and price

```
<response>
  <lst name="responseHeader">
    <int name="status">0</int>
    <int name="QTime">1</int>
  </lst>
  <result name="response" numFound="14" start="0">
    <doc>
      <arr name="cat">
        <str>electronics</str>
        <str>connector</str>
      </arr>
...
```

Querying Data: Results

- **The standard response format is JSON**


```
{
  "responseHeader":{
    "zkConnected":true,
    "status":0,
    "QTime":8,
    "params":{
      "q":"foundation"}},
  "response":{"numFound":4,"start":0,"maxScore":2.7879646,"docs":[
      {
        "id":"0553293354",
        "cat":["book"],
        "name":"Foundation",
        "price":7.99,
        "price_c":"7.99,USD",
        "inStock":true,
        "author":"Isaac Asimov",
        "author_s":"Isaac Asimov",
        "series_t":"Foundation Novels",
        "sequence_i":1,
        "genre_s":"scifi",
        "_version_":1574100232473411586,
        "price_c____l_ns":799}]
}}
```

Query Types Supported by Solr

- Single and multi-term queries
  - ex fieldname:value or title: software engineer
- +, -, AND, OR, NOT operators are supported
  - ex. title: (software AND engineer)
- Range queries on date or numeric fields,
  - ex: timestamp: [* TO NOW] or price: [1 TO 100]
- Boost queries:
  - e.g. title:Engineer ^1.5 OR text:Engineer
- Fuzzy search : is a search for words that are similar in spelling
  - e.g. roam~0.8 => noam
- Proximity Search : with a sloppy phrase query. The closer together the two terms appear, higher the score.
  - ex “apache lucene”~20 : will look for all documents where “apache” word occurs within 20 words of “lucene”

Copyright Ellis Horowitz, 2011-2022

Solr/Lucene is Used WorldWide

- Search Engine
  - Yandex.ru, DuckDuckGo.com
- Newspaper
  - Guardian.co.uk
- Music/Movies
  - Apple.com, Netflix.com
- Events
  - Stubhub.com, Eventbrite.com
- Cloud Log Management
  - Loggly.com
- Others
  - Whitehouse.gov
- Jobs
  - Indeed.com, Simplyhired.com, Naukri.com
- Auto
  - AOL.com
- Travel
  - Cleartrip.com
- Social Network
  - Twitter.com, LinkedIn.com, mylife.com

Tracking the popularity of Solr

According to DB-Engines Elasticsearch is ranked number one, and Solr is ranked number three

Copyright Ellis Horowitz, 2011-2022

Solr Includes Spell Checking

(you will implement this for homework #5)

- **Not enabled by default, see example config to wire it in**
- https://lucene.apache.org/solr/guide/7_7/spell-checking.html
- **Uses file or index-based dictionaries for spell correction**
- **Supports pluggable distance algorithms:**
  - **Levenstein alg:** https://en.wikipedia.org/wiki/Levenshtein_distance
  - **JaroWinkler alg: ,** https://en.wikipedia.org/wiki/Jaro%E2%80%93Winkler_distance
- **https://cwiki.apache.org/confluence/display/solr/SpellCheckComponent is a full discussion of the spellchecking abilities of Solr**

Copyright Ellis Horowitz, 2011-2022

Solr Includes Autosuggestion

(you will implement this for homework #5)

https://lucene.apache.org/solr/guide/7_7/suggester.html

Copyright Ellis Horowitz, 2011-2022

HW#4 Exercise

There are several components to this homework; you will possibly not need all of them;

1. Installing Ubuntu on Windows
2. Installing Solr
3. Using Solr to Index a web site
4. The actual exercise - comparing ranking algorithms
5. What to submit
6. Grading guidelines

Copyright Ellis Horowitz, 2011-2022

What you will need to do SUMMARY (1 of 2)

1. If you are using Windows, you will have to install Ubuntu on your machine
   - if you have a Mac, skip to step 2
   - Solr requires a Unix platform
   - first you must download and install Oracle's VirtualBox, free software that will permit you to install Ubuntu on your Windows device
   - second you will download and install Ubuntu within Virtualbox
2. Download and install the current release of Solr
   - there is a Quick Start Guide to help you at https://lucene.apache.org/solr/guide/7_7/solr-tutorial.html
   - the Solr server should be started; to check that it is running look at
   - http://localhost:8983/solr
3. Download the reference news website you are responsible for
   - 3 news websites are located on Google drive and are accessible using
   - https://drive.google.com/drive/folders/14qtJywCRk00wEbmt-Pv3efyPIBqwcoZr

Copyright Ellis Horowitz, 2011-2022

What you will need to do SUMMARY (2 of 2)

4. Import the news website web pages you are responsible for into Solr
   - There are only three new websites, use last digits of your USC ID to determine which one you are responsible for
5. Using your laptop install or use an existing web server to create a website
   - Macs (I believe) have Apache pre-installed
   - in your website create a web page that looks like a Google query box
   - in your website write a program that takes a query and sends it to Solr
   - in your website write a program that accepts the results from Solr and displays them as a web page
6. Run the set of 9 queries at the end of SolrExercise.pdf
   - save the top ten results for each of the 9 queries
7. Use the NetworkX library, the downloaded web pages and Jsoup library to create a network graph of the downloaded web pages
   - use the PageRank function from NetworkX to create a file, pagerankFile.txt that contains individual page ranks for every downloaded web page
   - install the pagerankFile in Solr and direct Solr to use it on queries
8. Run the set of 9 queries at the end of SolrExercise.pdf
   - save the top ten results for each of the queries
9. compare the two sets of results and describe any overlap

Copyright Ellis Horowitz, 2014-2023

Step 1: Ubuntu with VirtualBox

- **VirtualBox is an open source, freely available Windows application (it also runs on other platforms) that lets you run multiple operating systems on your single machine**
  - E.g. run Windows on a Mac, run Linux on Windows
  - Major supported operating systems include: Windows NT 4.0, Windows 2000, Windows 8, Windows 10, DOS Windows 3.x, Linux, Solaris, FreeBSD, OpenBSD
- **Ubuntu is a Linux-based operating system distributed on personal computers, smartphones and network servers. It uses Unity as its default desktop environment**
- **Solr requires a Unix environment to run, so step 1 is required if you plan to use your Windows laptop**

Copyright Ellis Horowitz, 2011-2022

Step 1: Setting Up Ubuntu with VirtualBox

1. Download the free version of VirtualBox for Windows machines
   - Instructions can be found here

   http://csci572.com/2022Fall/hw4/UbuntuVirtualBoxFinal.pdf
2. Download the Ubuntu 64-bit version
3. Run VirtualBox and select your Ubuntu version as the New Application
4. Set various parameters
5. Install Ubuntu and you should be ready to run

Copyright Ellis Horowitz, 2011-2022

Your Ubuntu Desktop

Copyright Ellis Horowitz 2011-2022

Step 2: Installing Solr

- **Solr is an open source enterprise search server based on the Lucene Java search library**
- **Instructions for downloading and installing Solr can be found here**
  - **https://solr.apache.org/guide/7_7/installing-solr.html**
- **Lucene is a fast, high performance, scalable search/information retrieval library**
  - Initially developed by Doug Cutting (Also author of Hadoop)
  - it provides for Indexing and Searching of documents
  - produces an Inverted Index of documents
  - Provides advanced Search options like synonyms, stopwords, based on similarity, proximity.

Copyright Ellis Horowitz, 2011-2022

Downloading Your Data Set of Web Pages

- We have created a reference set of web pages for all news websites and placed them on Google drive
- Below is a URL that will give you access to these data sets
  - Each data set contains three files
- The folders are:
  - NYTIMES [01 ~ 30]
  - FOXNEWS [31 ~ 60]
  - LATIMES [61 ~ 00]
- Download only the folder you are responsible for
  - Check your USC ID against the list in the exercise
- here is the URL again
  - https://drive.google.com/drive/folders/14qtJywCRk00wEbmt-Pv3efyPIBqwcoZr

Copyright Ellis Horowitz, 2011-2022

Step 3: Use Solr to Index a Web Site

- start the Solr server
- start a new Solr core
- Use Tika to import your saved files
- Use the Solr interface to check that the files have been properly indexed
- Note the URL:

localhost:8983/solr/#/myexample

- 19193 docs successfully

indexed

Copyright Ellis Horowitz, 2011-2022

Step 4: Comparing Search Engine Ranking Algorithms

1. You should download the reference files for the news website you are responsible for;
2. You should install Solr as described previously;
3. Take the web pages from the reference files and index them in Solr, as described in earlier slides
4. Build a front end to Solr that permits a visitor to enter a query and get matching results
5. Solr will return the results in JSON format; your server needs to take the results and format them for the user

Web Browser | Your Web Server | Your version of Solr

Copyright Ellis Horowitz, 2011-2022

Apache Solr Client APIs

- **http is the protocol to be used between client applications and Solr**
- **Clients use Solr's five basic operations: query, index, delete, commit, and optimize**
- **JavaScript is the standard client API and there is no package to be installed**
  - http requests can be sent using XMLHttpRequest
  - Solr will respond with JSON output
- **there is an output format specifically for Python**
  - See https://lucene.apache.org/solr/guide/7_7/using-python.html
- **Full list of Solr API clients can be found here**
  - https://lucene.apache.org/solr/guide/7_7/client-api-lineup.html

Copyright Ellis Horowitz, 2011-2022

Step 4

Here is a PHP client that accepts input from the user in a HTML form, and sends the request to the Solr server. After the Solr server processes the query, it returns the results which is parsed by the PHP program and formatted for display

```
<?php
// make sure browsers see this page as utf-8 encoded HTML
header('Content-Type: text/html; charset=utf-8');
$limit = 10;
$query = isset($_REQUEST['q']) ? $_REQUEST['q'] : false;
$results = false;
if ($query)
{  require_once('Apache/Solr/Service.php');
 // create a new solr service instance - host, port, and corename
 // path (all defaults in this example)
 $solr = new Apache_Solr_Service('localhost', 8983, '/solr/core_name/');
 // if magic quotes is enabled then stripslashes will be needed
 if (get_magic_quotes_gpc() == 1)
 {    $query = stripslashes($query);   }
```

returning a web page

test for a query

this is the Solr client library

Solr runs on port 8983

handles quoting of special characters in query

Copyright Ellis Horowitz, 2011-2022

PhP Program (2 of 3)

```
try
{   $results = $solr->search($query, 0, $limit);   }
catch (Exception $e)
{  die("<html><head><title>SEARCH EXCEPTION</title><body><pre>{$e->__toString()}</pre></body></html>");   }   }
?>
<html>  <head>  <title>PHP Solr Client Example</title>  </head>  <body>
 <form accept-charset="utf-8" method="get">
 <label for="q">Search:</label>
 <input id="q" name="q" type="text" value="<?php echo htmlspecialchars($query, ENT_QUOTES, 'utf-8'); ?>"/>
 <input type="submit"/>
 </form>
<?php
// display results
if ($results)
{  $total = (int) $results->response->numFound; $start = min(1, $total); $end = min($limit, $total);   ?>
 <div>Results <?php echo $start; ?> - <?php echo $end;?> of <?php echo $total; ?>:</div>
 <ol>
```

send query to Solr
catch any exception

create web page output

create input text box

create submit button

end form

JSON result string

Copyright Ellis Horowitz, 2011-2022

PhP Program (3 of 3)

```
<?php
 // iterate result documents
 foreach ($results->response->docs as $doc)
 {  ?>  <li>
 <table style="border: 1px solid black; text-align: left">
<?php
 // iterate document fields / values
 foreach ($doc as $field => $value)
 {   ?>
 <tr>
 <th><?php echo htmlspecialchars($field, ENT_NOQUOTES, 'utf-8'); ?></th>
 <td><?php echo htmlspecialchars($value, ENT_NOQUOTES, 'utf-8'); ?></td>
 </tr>
<?php   }
?> </table>  </li>
<?php   }   ?>
 </ol>
<?php
}  ?> </body>  </html>
```

output consists of a set of field, value pairs

Copyright Ellis Horowitz, 2011-2022

Comparing Ranking Algorithms

- **we have already seen the built-in Solr ranking method**
  - Based on Lucene
- **Solr permits alternative ranking algorithms**
  - to use PageRank we must create pageranks for all web pages in our news website
  - to do this we use an open source program called NetworkX
  - we need to create the web graph (pages and links) so the PageRank can be determined
  - See the following document for a method on extracting links and producing the NetworkX file:
  - http://csci572.com/2022Spring/hw4/ExtractingLinks.pdf

Copyright Ellis Horowitz, 2011-2022

http://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.link_analysis.pagerank_alg.pagerank.html

- You are going to use an open source PageRank algorithm, networkx, located at URL above;
- networkx is a Python program
- You need to create a graph that the PageRank algorithm can work on

Important Parameters:
- a NetworkX graph
- a damping parameter (e.g. 0.85)
- maximum number of iterations
- error tolerance
- starting Page Rank value of nodes

- (see expansion on next slide)

Source code for networkx.algorithms.link_analysis.pagerank_alg

http://networkx.github.io/documentation/networkx-1.10/_modules/networkx/algorithms/link_analysis/pagerank_alg.html#pagerank

```
def pagerank(G, alpha=0.85, personalization=None,
        max_iter=100, tol=1.0e-6, nstart=None, weight='weight',   dangling=None):
```

Return the PageRank of the nodes in the graph.

Parameters

G : *graph*: A NetworkX graph.

alpha : float, optional Damping parameter for PageRank, *default*=0.85.

personalization: dict, optional; By *default*, a uniform distribution is used.

max_iter : integer, optional Maximum number of iterations in power method eigenvalue solver.

tol : float, optional; Error tolerance used to check convergence in power method solver.

nstart : dictionary, optional Starting value of PageRank iteration for each node.

weight : key, optional; If None weights are set to 1.

Returns

pagerank : dictionary; Dictionary of nodes with PageRank as value

Examples

```
>>> G = nx.DiGraph(nx.path_graph(4))
>>> pr = nx.pagerank(G, alpha=0.9)
```

Copyright Ellis Horowitz, 2011-2022

Some NetworkX Operations

- Create an empty graph with no nodes and no edges.

```
>>> import networkx as nx
>>> G=nx.Graph()
```

- add one node at a time

```
>>>G.add_node(1)
```

- add a list of nodes

```
>>>G.add_nodes_from([2,3])
```

- add one edge at a time

```
>>>G.add_edge(1,2)
>>>e=(2,3)
>>>G.add_edge(*e) #unpack edge tuple*
```

- add a list of edges

```
>>>G.add_edges_from([1,2),(1,3)])
```

Copyright Ellis Horowitz, 2011-2022

Final Steps

- **Input to the PageRank algorithm a file containing every document ID and associated with each ID, the IDs that are pointed to by links within the document ID**
- **Output from the PageRank algorithm is a file containing every document ID and its associated PageRank**
- **place this file in solr-7.7.3/server/solr/core_name, call the file external_pageRankFile.txt**
- **add the PageRank field to the schema.xml file**

```
<fieldType name="external" keyField="id" defVal="0" class="solr.ExternalFileField" />
<field name="pageRankFile" type="external" stored="false" indexed="false" />
```

Copyright Ellis Horowitz, 2011-2022

The Queries

- **Once both ranking algorithms are working you should input the queries below and compare the results**

Nine Queries
Cannes
Congress
Democrats
Patriot Movement
Republicans
Senate
Olympics 2020
Stock
Virus

Copyright Ellis Horowitz, 2011-2022

Spell Checking and Correction

Some Google/Bing Examples

Russian mathematician, notice red underline appears as soon as the first incorrect character is typed

Bing also combines autocomplete with spelling correction but there is no red underline

Himilayas

Copyright Ellis Horowitz 2011-2022

More Examples

easy for people, but harder for computers to correct, **likely use of n-grams**

easy for a computer to correct **likely use of a database** of words

computer needs to both identify the error and correct the misspelling

Google combines spelling correction with the **most likely terms** as it comes up with "cars" in autocomplete for the query "jagwa" (misspelled) but leaves the user's misspelling for a while

Copyright Ellis Horowitz 2011-2022

Spelling Correction is Done in Many Places

1. Word processing

2. Smartphone input

Word processing is the classic application for spelling correction

Word and PowerPoint have mode to auto-correct

- set as the default
- the spell dictionary can be modified

Typing on a virtual keyboard can be doubly difficult (for seniors)

Copyright Ellis Horowitz, 2011-2022

Rates of Spelling Errors

Error rates vary depending upon the application

- **Typing is very error prone, and especially difficult on smartphones**
- **Different studies have produced varying results**

26%: Web queries Wang *et al.* 2003

13%: Retyping, no backspace: Whitelaw *et al.* English&German

7%: Words corrected retyping on phone-sized organizer

So seamless spelling correction is an essential component of information retrieval and especially for search engines

Copyright Ellis Horowitz, 2011-2022

The Two Main Spelling Tasks

1. **Spelling Error *Detection***
   - Obviously we need a big dictionary and the ability to search it quickly
   - Using context may be necessary
     - To do this spelling error models have been devised
2. **Spelling Error *Correction***
   - Web search engines **always** try to suggest a correction
   - Autocomplete requires spelling correctors to anticipate the final word
     - Fast response time is required
   - The two major techniques are
     1. edit distance algorithms or
     2. n-gram matching to come up with the correction

Copyright Ellis Horowitz, 2011-2022

Three Types of Spelling Errors

1. **Non-word errors**
   - *graffe* → *giraffe*
2. **Typographical errors** (flip the *e* and *r*)
   - *three* → *there* *(especially bad as both are legal words)*
3. **Cognitive errors** (homophones, sounds alike)
   - *piece* → *peace*,
   - *too* → *two*
   - *your* → *you're*

Copyright Ellis Horowitz, 2011-2022

Non-Word Spelling Errors

- **Non-word spelling error detection:**
  - Any word not in a *dictionary* is presumed to be an error
  - The larger the dictionary the better
- **Approach to non-word spelling error correction:**
  - Generate candidates from the dictionary that are close to the error
  - **How do we do this?**
    - **Shortest weighted edit distance**
      - **Edit distance** is a way of quantifying how dissimilar two strings (e.g., words) are to one another by counting the minimum number of operations required to transform one string into the other
    - **Highest noisy channel probability**
      - use probabilities to select the most obvious candidates

Copyright Ellis Horowitz, 2011-2022

Causes of Misspellings

Cause	Misspelling	Correction
typing quickly	exxit mispell	exit misspell
keyboard adjacency	importamt	important
inconsistent rules	conceive concierge	conceive concierge
ambiguous word breaking	silver light	silverlight
new words	kinnect	kinect

e before i
i before e

According to Cucerzan and Brill, **more than 10% of search engine queries are misspelled**
"*Spelling Correction as an iterative process that exploits the collective knowledge of web users*"
http://csci572.com/papers/Cucerzan.pdf (advocates using query logs to guess the correct spelling)

Copyright Ellis Horowitz, 2011-2022

How Many Error Variations of a Word May Actually Appear

http://www.netpaths.net/blog/britney-spears-spelling-variations/

Britney Spears spelling co

www.google.com/jobs/archive/britney.html

Apps | CSCI 571 Home Page | CSCI 572 Home Page | Computer Science D... | PHP & MySQL | ITS - Software | GMail | Apple | Other bookmarks

The data below shows some of the misspellings detected by our spelling correction system for the query [britney spears], and the count of how many different users spelled her name that way. Each of these variations was entered by at least two different unique users within a three month period, and was corrected to [britney spears] by our spelling correction system (data for the correctly spelled query is shown for comparison).

Return to Google's jobs pages

488941 britney spears	29 britent spears	9 brinttany spears	5 brney spears	3 britiy spears	2 brirreny spears
40134 brittany spears	29 brittnany spears	9 britanay spears	5 broitney spears	3 britmeny spears	2 brirtany spears
36315 brittney spears	29 britttany spears	9 britinany spears	5 brotny spears	3 britneeey spears	2 brirttany spears
24342 britany spears	29 btiney spears	9 britn spears	5 bruteny spears	3 britnehy spears	2 brirttney spears
7331 britny spears	26 birttney spears	9 britnew spears	5 btiyney spears	3 britnely spears	2 britain spears
6633 briteny spears	26 breitney spears	9 britneyn spears	5 btrittney spears	3 britnesy spears	2 britane spears
2696 britteny spears	26 brinity spears	9 britrney spears	5 gritney spears	3 britnetty spears	2 britaneny spears
1807 briney spears	26 britenay spears	9 britny spears	5 spritney spears	3 britnex spears	2 britania spears
1635 brittny spears	26 britneyt spears	9 brtittney spears	4 bittny spears	3 britneyxxx spears	2 britann spears
1479 brintey spears	26 brittan spears	9 brtny spears	4 bnritney spears	3 britnity spears	2 britanna spears
1479 britanny spears	26 brittne spears	9 brytny spears	4 brandy spears	3 britntey spears	2 britannie spears
1338 britiny spears	26 bttitany spears	9 rbitney spears	4 brbritney spears	3 britnyey spears	2 britannt spears
1211 britnet spears	24 beitney spears	8 birtiny spears	4 breatiny spears	3 britterny spears	2 britannu spears
1096 britiney spears	24 birteny spears	8 bithney spears	4 breetney spears	3 brittneey spears	2 britanyl spears
991 britaney spears	24 brightney spears	8 brattany spears	4 bretiney spears	3 brittnney spears	2 britanyt spears
991 britnay spears	24 brintiny spears	8 breitny spears	4 brfitney spears	3 brittnyey spears	2 briteeny spears
811 brittney spears	24 britanty spears	8 breteny spears	4 briattany spears	3 brityen spears	2 britenany spears
811 brtiney spears	24 britenny spears	8 brightny spears	4 brieteny spears	3 briytney spears	2 britenet spears
664 birtney spears	24 britini spears	8 brintay spears	4 briety spears	3 brltney spears	2 briteniy spears
664 brintney spears	24 britnuy spears	8 brinttey spears	4 briitny spears	3 broteny spears	2 britenys spears
664 briteney spears	24 brittni spears	8 briotney spears	4 briittany spears	3 brtaney spears	2 britianey spears
601 bitney spears	24 brittnie spears	8 britanys spears	4 brinie spears	3 brtiiany spears	2 britin spears
601 brinty spears	21 biritney spears	8 britley spears	4 brinteney spears	3 brtinay spears	2 britinary spears
544 brittaney spears	21 birtany spears	8 britneyb spears	4 brintne spears	3 brtinney spears	2 britmy spears
544 brittnay spears	21 biteny spears	8 britnrey spears	4 britaby spears	3 brtitany spears	2 britnaney spears
364 britey spears	21 bratney spears	8 britnty spears	4 britaey spears	3 brtiteny spears	2 britnat spears
364 britiny spears	21 britani spears	8 brittner spears	4 britainey spears	3 brtnet spears	2 britnbey spears
329 brtney spears	21 britanie spears	8 brottany spears	4 britinie spears	3 brytiny spears	2 britndy spears
269 bretney spears	21 briteany spears	7 baritney spears	4 britinney spears	3 btney spears	2 britneh spears
269 britneys spears	21 brittay spears	7 birntey spears	4 britmey spears	3 drittney spears	2 britneney spears
244 britne spears	21 brittinay spears	7 biteney spears	4 britnear spears	3 pretney spears	2 britney6 spears
244 brytney spears	21 brtany spears	7 bitiny spears	4 britnel spears	3 rbritney spears	2 britneye spears
220 breatney spears	21 brtiany spears	7 breateny spears	4 britneuy spears	2 barittany spears	2 britneyh spears
220 britiany spears	19 birney spears	7 brianty spears	4 britnewy spears	2 bbbritney spears	2 britneym spears
199 britnney spears	19 brirtney spears	7 brintye spears	4 britnney spears	2 bbitney spears	2 britneyyy spears
163 britnny spears	19 britnaey spears	7 britianny spears	4 brittaby spears	2 britny spears	2 britnhey spears
147 breatny spears	19 brittee spears	7 britly spears	4 brittery spears	2 bbrittany spears	2 britnjey spears
147 brittiney spears	19 britony spears	7 britnej spears	4 britthey spears	2 beitany spears	2 britnne spears
147 britty spears	19 brittanty spears	7 britneyu spears	4 brittnaey spears	2 beitny spears	2 britnu spears
147 brotney spears	19 britttney spears	7 britniey spears	4 brittnat spears	2 bertney spears	2 britoney spears
147 brutney spears	17 birtny spears	7 britnnay spears	4 brittneny spears	2 bertny spears	2 britrany spears
133 britteney spears	17 brieny spears	7 brittian spears	4 brittnye spears	2 betney spears	2 britreny spears
133 briyney spears	17 brintty spears	7 briyny spears	4 brittteny spears	2 betny spears	2 britry spears
121 bittany spears	17 brithy spears	7 brrittany spears	4 briutney spears	2 bhriney spears	2 britsany spears
121 bridney spears	17 brittanie spears	7 brttiney spears	4 briyeny spears	2 biney spears	2 brittanay spears
121 britainy spears	15 brinney spears	7 btiteny spears	4 brnity spears	2 bintey spears	2 brittang spears
121 britmey spears	15 briten spears	7 btrittany spears	4 brtteny spears	2 biretny spears	2 brittans spears
109 brietney spears	15 briterney spears	6 beritny spears	4 brttiany spears	2 biritany spears	2 brittanyh spears
109 brithny spears	15 britheny spears	6 bhritney spears	4 bryney spears	2 birittany spears	2 brittanyn spears
109 britni spears	15 britneny spears	6 birthney spears	4 brythney spears	2 birittny spears	2 brittany's spears
109 brittant spears	15 brittamy spears	6 breathney spears	4 brytne spears	2 birnty spears	2 brittanyt spears
98 bittney spears	15 brittmey spears	6 breaty spears	4 brytni spears	2 birtey spears	2 brittanyy spears
98 brithey spears	15 brytnei spears	6 bretany spears	4 brytnie spears	2 britheny spears	2 brittary spears
98 brittiany spears	15 btriney spears	6 briatany spears	4 bvritney spears	2 birtiney spears	2 brittenie spears
98 btitney spears	15 rittney spears	6 brint spears	4 dritney spears	2 birtnay spears	2 brittenty spears
89 brietny spears	14 brinet spears	6 britenney spears	4 priteny spears	2 birtnet spears	2 brittinney spears
89 brinety spears	14 britneyy spears	6 britian spears	3 beittany spears	2 bitnet spears	2 brittley spears
89 brintny spears	14 britten spears	6 britinty spears	3 bichney spears	2 bitttany spears	2 brittn spears
89 britnie spears	12 beritney spears	6 brititney spears	3 biritny spears	2 bnrittany spears	2 brittnery spears
89 brittey spears	12 bretiny spears	6 britnsy spears	3 birnety spears	2 bntney spears	2 brittnety spears
89 brittnet spears	12 briatny spears	6 britrey spears	3 birny spears	2 bpitney spears	2 brittney's spears
89 brity spears	12 brieney spears	6 britzny spears	3 birteney spears	2 brandi spears	2 brittneyu spears
89 ritney spears	12 brinany spears	6 brittine spears	3 britthy spears	2 brattney spears	2 brittnsey spears
80 bretny spears	12 britaany spears	6 brittnry spears	3 birtiney spears	2 breanty spears	2 britttaney spears

Google's list of spelling errors for "Britney Spears" includes a count of how many different users spelled her name in various way, e.g. 488,941 people used the correct spelling "britney spears" while 40,134 used "britnay spears"

there are 593 different variations that actually occurred

Spelling Errors Needing Context

- **Some misspellings require context to disambiguate**
  1. **consider whether the surrounding words "make sense" for your candidate set, e.g.**
     - *Flying form Heathrow to LAX → Flying from Heathrow to LAX*
     - *Power crd → power cord versus Video crd → video card*
  2. **For candidate words with similar sounds but different spellings and different meanings, context is needed**
     - *e.g. there, their*
     - *N-grams are most useful here*
  3. **To resolve the above, candidate words must be found with similar *pronunciations***
     - *use the Soundex algorithm*

Copyright Ellis Horowitz, 2011-2022

More Challenges for Identifying Spelling Errors

- **Some additional challenges**
  4. **Allow for insertion of a space or hyphen**
     - `thisidea → this idea`
     - `inlaw → in-law`
     - `chat inspanich → chat in spanish` (2 corrections)
  5. **Allow for deletion of a space**
     - `power point slides → powerpoint slides`
  6. **Watch out for words NOT in the Lexicon that are valid, e.g.**
     - amd processors
       - AMD is a company that makes processors for laptops
       - Another example where context is needed

Copyright Ellis Horowitz, 2011-2022

The Noisy Channel Model

- This model suggests treating the misspelled word as if a correctly spelled word has been distorted by being passed through a noisy communication channel
- Noise in this case refers to substitutions, insertions or deletions of letters

Copyright Eric Horowitz, 2011-2022

Bayesian Inference Implies We Can Use Previous Combinations to Predict the Correct Word

- **We see an observation *x* (a misspelled word) and our job is to find the correct word *w* that generated this misspelled word**
- **Out of all possible words in the vocabulary *V* we want to find the word *w* such that the probability that w is the correct word, *P(w|x)*, is highest. We use the hat notation ˆ to mean "our estimate of the correct word". We state this in the following way:**

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} P(w|x)$$

- **Out of all words in the vocabulary, we want the particular word that maximizes the right-hand side *P(w|x)***
- ***Bayes Rule:***

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)}$$

The probability of *a* given *b* is equal to the probability of *b* given *a*, times the probability of *a*, divided by the probability of *b*

implies $\hat{w} = \underset{w \in V}{\operatorname{argmax}} \frac{P(x|w)P(w)}{P(x)}$ and simplifying gives $\hat{w} = \underset{w \in V}{\operatorname{argmax}} P(x|w)\,P(w)$

Note: *P(x|w)* is a constant so we can ignore it

Conclusion: we need a table of probabilities

Copyright Ellis Horowitz, 2011-2022

The Basic Spelling Correction Algorithm

1. **Initial step**: Create a dictionary and encode it for fast retrieval
2. When a query is submitted, the spell checker examines each word and for words not in the dictionary looks for possible character edits, namely
   - insertions,
   - deletions,
   - substitutions, and occasionally
   - transpositions
   - **Observation:**
     - 80% of errors are within edit distance 1
     - Almost all errors within edit distance 2
3. Take the output of step 2 and compute probabilities for the candidates using previously identified probability tables created from n-grams
4. Select the result with highest probability

Copyright Ellis Horowitz, 2011-2022

The Basic Spelling Correction Algorithm Refined

- **Edit distance** is a way of quantifying how dissimilar two strings (e.g. words) are to one another by counting the minimum number of operations required to transform one string into the other
  - different algorithms assume slightly different operations
  - e.g., Levenshtein uses: removal, insertion, substitution of a character

1. **Check each query term against the dictionary**
2. **For each term NOT found in the dictionary, generate all words within edit distance ≤ *k* (e.g., *k* = 1 or 2) and then intersect them with dictionary**
   - Compute them fast with a Levenshtein algorithm
3. **Use a character *n*-gram index and find dictionary words that share "most" *n*-grams with word**
4. **Select the word with the highest probability** (largest n-gram count)
5. **For speed, have a pre-computed map of words to possible corrections**

Copyright Ellis Horowitz, 2011-2022

Use Edit Distance To Produce Candidate Corrections

Input	Candidate Correction	Correct Letter	Error Letter	correction Type
acress	actress	t	–	insertion
acress	cress	–	a	deletion
acress	caress	ca	ac	transposition
acress	access	c	r	substitution
acress	across	o	e	substitution
acress	acres	–	s	deletion

Six words within 1 of acress

Context check is necessary to choose the appropriate word

(try this yourself in Google/Bing)

For the word "acress" there are six dictionary words all within edit distance 1

Copyright Ellis Horowitz, 2011-2022

Now Apply Probabilities

- We now need to compute the prior probability of each occurrence
- We can do this using unigrams, bigrams, trigrams, etc
- Using the Corpus of Contemporary English, 404,253,213 words we get the following
- *Across* is the most likely choice, followed by

For fun try:
I need acress to . . .
I love the acress . . .
a kiss and a acress . . .

word	Frequency of word	*P(w)*
actress	9,321	.0000230573
cress	220	.0000005442
caress	686	.0000016969
access	37,038	.0000916207
across	120,844	.0002989314
acres	12,874	.0000318463

"***across***" is the most likely correction → across

Copyright Ellis Horowitz, 2011-2022

The Spelling Correction Dictionary and Autocomplete

- **The search for corrections is carried out from left-to-right**
  - At each point, a partial hypothesis is expanded with every character which could follow the partial hypothesis and lead to one of the known words (the user input is always allowed as an output hypothesis).
  - Thus the branching factor controls the amount of time required to search for spelling corrections.
- **The terms of the lexicon must be stored in a data structure that affords efficient *prefix matching***
  - Often a trie data structure is used

Copyright Ellis Horowitz, 2011-2022

The Spelling Correction Dictionary
Example of a Trie

- **a prefix tree (sometimes called a trie from the word retrieval) is a tree of degree >=2 in which the branching at any level is determined by a portion of the key**

branch nodes take you down the tree to element nodes; At any stage one is pointing at all keyword matches that contain the same prefix; Computing time for retrieval is *O(m)* where *m* is the length of the string, at the expense of increased storage

Copyright Ellis Horowitz, 2011-2022

The Spelling Correction Dictionary Error Test Sets

- **To enhance a lexicon one can include a table of common misspellings**
- **there are many possible spelling error test sets, e.g.**
  - **Wikipedia's list of common English misspelling**
    - https://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings
  - **Aspell filtered version of that list**
    - http://aspell.net/ is the spell program
  - **Birkbeck spelling error corpus**
    - https://ota.bodleian.ox.ac.uk/repository/xmlui/handle/20.500.12024/0643, 1.8MBs
- **These sets are primarily used to correct typographical errors**

Copyright Ellis Horowitz, 2011-2022

Using N-Grams For Spelling Correction

- An ***n*-gram model** is a type of probabilistic language model for predicting the next item in a sequence
- Two benefits of *n*-gram models (and algorithms that use them) are simplicity and scalability – with larger *n*, a model can store more context with a well-understood space–time tradeoff, enabling small experiments to scale up efficiently

- **Sample of 3-gram sequences**
- ceramics collectables fine (130)
- ceramics collected by (52)
- ceramics collectible pottery (50)
- ceramics collectibles cooking (45)

Sample of 4-gram sequences and # of times appeared

serve as the incoming (92)
serve as the incubator (99)
serve as the independent (794)
serve as the index (223)
serve as the indication (72)
serve as the indicator (120)

a query such as "serve as the indapendant" would match the above

Copyright Ellis Horowitz, 2011-2022

Google's N-Gram Data

- Google has collected and uses a great deal of N-gram data
- Google is using the Linguistics Data Consortium to distribute more than one trillion words they have extracted from public web pages
- Below is a statistical summary of the data they are distributing

File sizes: approx. 24 GB compressed (gzip'ed) text files

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

http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html

Copyright Ellis Horowitz 2011-2022

Handling N-Gram Data

- **Efficiency considerations are important when building language models that use such large sets of N-grams.**
- **Rather than store each word as a string, it is generally represented in memory as a 64-bit hash number, with the words themselves stored on disk.**
- **Probabilities are generally quantized using only 4-8 bits (instead of 8-byte floats), and N-grams are stored in reverse tries.**
- **N-grams can also be shrunk by pruning, for example only storing N-grams with counts greater than some threshold (such as the count threshold of 40 used for the Google N-gram release)**

- See *Handling Massive N-Gram Datasets Efficiently*, G. Pibiri and R. Venturini which improves upon *Elias-Fano encoding*, https://www.antoniomallia.it/sorted-integers-compression-with-elias-fano-encoding.html

Copyright Ellis Horowitz, 2011-2022

Applying the N-Gram Model to Spelling Correction

- Let's say you are using 4-grams to calculate the probability of a next word in text.
  - You have "this is a very" followed by "sunny".
  - But assume "sunny" does not occur in the n-gram set that includes "this is a very"
  - *Implication*: so for the 4-gram model "sunny" has probability 0,

- Back-off means you go back to a *n-1*-gram level to calculate the probabilities when you encounter a word with probability = 0.
  - So in the above you will use a 3-gram model to calculate the probability of "sunny" in the context "is a very".

- Whenever you go back one level you multiply the odds by an empirically derived number, in this case 0.4. So if sunny exists in the 3-gram model the probability would be
  *0.4 * P("sunny"|"is a very")*

Generalizing, this leads to an algorithm for handling n-grams to do spelling correction, called *The Stupid Back-off Algorithm*

1. If a higher-order N-gram has a zero count, we simply backoff to a lower order N-gram, weighed by a fixed (context-independent) weight
2. The backoff terminates in the unigram

- **See Wikipedia's description of the Back-off Model**

Copyright Ellis Horowitz, 2011-2022

A Complete Spelling Correction Program

Copyright Ellis Horowitz 2011-2022

Peter Norvig's Spelling Corrector written in Python

```
import re, collections
def words(text): return re.findall('[a-z]+', text.lower())
def train(features):
    model = collections.defaultdict(lambda: 1)
    for f in features:
        model[f] += 1
    return model
NWORDS = train(words(file('big.txt').read()))
alphabet = 'abcdefghijklmnopqrstuvwxyz'
def edits1(word):
   splits     = [(word[:i], word[i:]) for i in range(len(word) + 1)]
   deletes    = [a + b[1:] for a, b in splits if b]
   transposes = [a + b[1] + b[0] + b[2:] for a, b in splits if len(b)>1]
   replaces   = [a + c + b[1:] for a, b in splits for c in alphabet if b]
   inserts    = [a + c + b     for a, b in splits for c in alphabet]
   return set(deletes + transposes + replaces + inserts)

def known_edits2(word):
    return set(e2 for e1 in edits1(word) for e2 in edits1(e1) if e2 in NWORDS)

def known(words): return set(w for w in words if w in NWORDS)

def correct(word):
    candidates = known([word]) or known(edits1(word)) or known_edits2(word) or [word]
    return max(candidates, key=NWORDS.get)
```

Correct takes a word and returns a likely correct form of the word:

```
correct('speling')
Spelling
correct('korrecter')
corrector
```

http://norvig.com/spell-correct.html

Copyright Ellis Horowitz, 2011-2022

How the Program Works

- The algorithm and its description can be found at
- http://norvig.com/spell-correct.html
- The file big.txt contains a million words
  - It includes text of books from Project Gutenberg, Wiktionary, British National Corpus
- Extract the individual words (function `words` converts everything to lower case)
- *The* and *the* are the same, but "don't" is seen as "don" and "t"
- The program counts how many times each word occurs using function `train`
- `NWORDS[w]` holds a count of how many times the word w has been seen
- For words that are not in our set we set their occurrence to a default, non-zero value of 1 using the Python hash table statement `collections.defaultdict`
- Edit distance is the number of changes it would take to turn one word into another
- An edit can be one of: {deletion, transposition, alteration, insertion}
- The 7 lines that end in `return set(deletes + transposes + replaces + inserts)` is a function that returns a set of all words c that are one edit away from w
  - This can be a large set
- **The literature on spelling correction claims that 80-95% of spelling errors are an edit distance of 1 from the target**

Copyright Ellis Horowitz, 2011-2022

How the Program Works (cont'd)

- To compute edit distance 2, just apply edits1 to all the results of edits1

```
Def edit2(word):
    return set(e2 for e1 in edits1(word) for e2 in edits1(e1))
```

- The function known_edits2 removes words that are not in our original set and greatly reduces the size of edits2
  - E.g. all edit 2 combinations for "something" produces 114,324 words, but pruning leaves only 4 words: smoothing, seething, something, and soothing
- Now he assumes his probability model, which is "all known words of edit distance 1 are infinitely more probable than known words of edit distance 2
- He defines function `correct` which chooses as the set of candidate words the set with the shortest edit distance to the original word as long as the set has some known words
- See his web page for an evaluation of his program

Copyright Ellis Horowitz, 2011-2022

Natural Language Corpus Data

- Peter Norvig has provided a web page that is full of useful data for a spelling corrector
  - http://norvig.com/spell-correct.html
- Peter Norvig's list of errors
- Peter Norvig's list of counts of single-edit errors
- http://norvig.com/ngrams/

Copyright Ellis Horowitz, 2011-2022

Some References

- ***How Difficult is it to Develop a Perfect Spell-checker? A Cross-linguistic Analysis through Complex Network Approach*, Monojit Choudhury1, Markose Thomas2, Animesh Mukherjee1, Anupam Basu1, and Niloy Ganguly1**

http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=52A3B869596656C9DA285DCE83A0339F?doi=10.1.1.146.4390&rep=rep1&type=pdf

- ***Using the web for language independent spellchecking and autocorrection* by C. Whitelaw et al Proc. 2009 Conf. on Empirical Methods in Natural Language Processing, pp890-899**

http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/pubs/archive/36180.pdf

Spell Checking by Computer, by Roger Mitton,

http://www.dcs.bbk.ac.uk/~roger/spellchecking.html

Copyright Ellis Horowitz, 2011-2022

Edit Distance & Levenshtein Algorithm

Copyright Ellis Horowitz 2011-2018

Edit Distance

- the minimum edit distance between two strings is the minimum number of editing operations
  - insertion
  - deletion
  - substitution

  needed to transform one into the other

Copyright Ellis Horowitz, 2011-2022

How to Find the Minimum Edit Distance

- Searching for a path (sequence of edits) from the start string to the final string
  - initial state: word we're transforming (e.g. kitten)
  - operators: insert, delete, substitute
  - goal state: the word we're trying to get to (e.g. sitting)
  - path cost: what we want to minimize, the number of edits
- If we blindly generate all possible paths in an effort to produce the goal state, our algorithm will take exponentially long
- But we realize that we needn't do that, we may only follow the path that is optimal at each stage

Optimal Solution
Sub "s" for "k"
Sub "i" for "e"
Ins "g" at the end

Copyright Ellis Horowitz, 2011-2022

Dynamic Programming for Minimum Edit Distance

- **Dynamic programming: a technique for solving optimization problems by combining solutions to sub-optimal problems bottom-up**
- **Given two strings: *X* of length *n* and *Y* of length *m*, define**
- ***d(i, j)* as**
  - **the minimum edit distance between *X[1..i]* and *Y[1..j]***
    - **i.e. the first *i* characters of *X* and the first *j* characters of *Y***
  - **Then the minimum edit distance between X and Y is thus *D(n,m)***
- **The dynamic programming algorithm will**
  - **compute *D(i,j)* for small *i, j***
  - **and compute larger *D(i,j)* based on previously computed smaller values**
  - **i.e. compute *D(i,j)* for *all i* $(0 < i < n)$ and *j* $(0 < j < m)$**

- **Note**: the dynamic programming technique was invented by Richard Bellman while at the Rand Corp, later a professor at USC

Copyright Ellis Horowitz, 2011-2022

Pseudocode Implementation of Levenshtein Distance

```
function LevenshteinDistance(char s[1..m], char t[1..n]):
//for all i and j, d[i,j] will hold the Levenshtein distance between
//the first i characters of s and the first j characters of t
declare int d[0..m, 0..n]
//Set each element in d to zero
for i from 1 to m, j from 1 to n: d[i,j] := 0
Starting with empty character source and target can be easily produced by inserts
for i from 1 to m: d[i,0] := i
for j from 1 to n: d[0,j] := j
//main loop
for j from 1 to n:
        for i from 1 to m:
                if s[i] = t[j] then substitutionCost := 0 else substitutionCost := 1
                d[i,j] := min (d[i-1,j] + 1,                    // deletion
                               d[i,j-1] + 1,                    //insertion
                              d[i-1, j-1] + substitutionCost  //substitution
                             )
return d[m,n]
```

Copyright Ellis Horowitz 2011-2022

Levenshtein Example

- **Consider the strings: sitting, kitten**
- **Here is the initial matrix**

	#	K	I	T	T	E	N
#	0	1	2	3	4	5	6
S	1						
I	2						
T	3						
T	4						
I	5						
N	6						
G	7						

is the null string
Looking at the first column, to go from # to S requires 1 insert; to go from # to SI requires 2 inserts; to go from # to SIT requires 3 inserts;
Similarly for the first row, to go from # to K requires 1 insert, etc.

Copyright Ellis Horowitz, 2011-2022

Levenshtein Allows For Insertion, Deletion and Substitution

- **To get the value in the 1,1 position use the formula:**

d(i , j) = min { d(i-1,j) + 1, d(i, j-1) + 1, d(i-1, j-1) + 1 if chars not equal, 0 otherwise}

We can fill in the second row and the second column observing that at each step there is a single deletion followed by insertions

	#	K	I	T	T	E	N
#	0	1	2	3	4	5	6
S	1	1	2	3	4	5	6
I	2	2	1	2	3	4	5
T	3	3	2	1	2	3	4
T	4	4	3	2	1	2	3
I	5	5	4	3	2	2	3
N	6	6	5	4	3	3	2
G	7	7	6	5	4	4	3

Copyright Ellis Horowitz, 2011-2022

Levenshtein Example 2

	#	S	A	T	U	R	D	A	Y
#	0	1	2	3	4	5	6	7	8
S	1	0	1	2	3	4	5	6	7
U	2	1	1	2	2				
N	3								
D	4								
A	5								
Y	6								

Copyright Ellis Horowitz 2011-2022

Performance

- **Time:** $O(nm)$
- **Space:** $O(nm)$
- **Backtrace:** $O(n + m)$

Copyright Ellis Horowitz, 2011-2022

Computing Alignments

- **Edit distance isn't sufficient**
  - we often need to align each character of the two strings to each other
- **we do this by keeping a "backtrace"**
- **every time we enter a cell, remember where we came from**
- **so when we reach the end,**
  - trace back the path from the upper right corner to read off the alignment
- For example, for each field of the table we can add a field,

ptr(i, j) whose value is either

  - LEFT, for insertion
  - DOWN, for deletion
  - DIAG, for substitution

Copyright Ellis Horowitz, 2011-2022

Weighted Edit Distance

- **why would we add weights to the computation?**
  - **spell correction: some letters are more likely to be mistyped than others**
- a **confusion matrix** is a specific table layout that allows visualization of the performance of an algorithm; each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class (or vice-versa)

X \ Y (correct)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	0	0	7	1	342	0	0	2	118	0	1	0	0	3	76	0	0	1	35	9	9	0	1	0	5	0
b	0	0	9	9	2	2	3	1	0	0	0	5	11	5	0	10	0	0	2	1	0	0	8	0	0	0
c	6	5	0	16	0	9	5	0	0	0	1	0	7	9	1	10	2	5	39	40	1	3	7	1	1	0
d	1	10	13	0	12	0	5	5	0	0	2	3	7	3	0	1	0	43	30	22	0	0	4	0	2	0
e	388	0	3	11	0	2	2	0	89	0	0	3	0	5	93	0	0	14	12	6	15	0	1	0	18	0
f	0	15	0	3	1	0	5	2	0	0	0	3	4	1	0	0	0	6	4	12	0	0	2	0	0	0
g	4	1	11	11	9	2	0	0	0	1	1	3	0	0	2	1	3	5	13	21	0	0	1	0	3	0
h	1	8	0	3	0	0	0	0	0	0	2	0	12	14	2	3	0	3	1	11	0	0	2	0	0	0
i	103	0	0	0	146	0	1	0	0	0	0	6	0	0	49	0	0	0	2	1	47	0	2	1	15	0
j	0	1	1	9	0	0	1	0	0	0	0	2	1	0	0	0	0	0	5	0	0	0	0	0	0	0
k	1	2	8	4	1	1	2	5	0	0	0	0	5	0	2	0	0	0	6	0	0	0	4	0	0	3
l	2	10	1	4	0	4	5	6	13	0	1	0	0	14	2	5	0	11	10	2	0	0	0	0	0	0
m	1	3	7	8	0	2	0	6	0	0	4	4	0	180	0	6	0	0	9	15	13	3	2	2	3	0
n	2	7	6	5	3	0	1	19	1	0	4	35	78	0	0	7	0	28	5	7	0	0	1	2	0	2
o	91	1	1	3	116	0	0	0	25	0	2	0	0	0	0	14	0	2	4	14	39	0	0	0	18	0
p	0	11	1	2	0	6	5	0	2	9	0	2	7	6	15	0	0	1	3	6	0	4	1	0	0	0
q	0	0	1	0	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
r	0	14	0	30	12	2	2	8	2	0	5	8	4	20	1	14	0	0	12	22	4	0	0	1	0	0
s	11	8	27	33	35	4	0	1	0	1	0	27	0	6	1	7	0	14	0	15	0	0	5	3	20	1
t	3	4	9	42	7	5	19	5	0	1	0	14	9	5	5	6	0	11	37	0	0	2	19	0	7	6
u	20	0	0	0	44	0	0	0	64	0	0	0	0	2	43	0	0	4	0	0	0	0	2	0	8	0
v	0	0	7	0	0	3	0	0	0	0	0	1	0	0	1	0	0	0	8	3	0	0	0	0	0	0
w	2	2	1	0	1	0	0	2	0	0	1	0	0	0	0	7	0	6	3	3	1	0	0	0	0	0
x	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
y	0	0	2	0	15	0	1	7	15	0	0	0	2	0	6	1	0	7	36	8	5	0	0	1	0	0
z	0	0	0	7	0	0	0	0	0	0	0	7	5	0	0	0	0	2	21	3	0	0	0	0	3	0

the confusion matrix for spelling errors shows us, e.g. that "e" is most often confused with "a",
and that "i" is often confused with both "e" and "a"

USC

USC Viterbi
School of Engineering

Snippets
Normal, Featured and Rich

Snippets in Google Search

USC Viterbi School of Engineering

In 1998, Google introduced *Snippet*, a short description of, or excerpt from, a website which appears in Google search results. Snippets are created automatically based on the site's content and the query terms

key query terms are highlighted:
number
x-
y-intercepts
quadratic
functions

notice that
have and **may** are also
in bold

USC

Some Elementary Facts About Google Snippets

- **if the snippet begins with ellipses (. . . .) that indicates the snippet was excerpted from a larger body of text and text preceding the ellipses was omitted**
- **when ellipses follow at the end of the snippet, the snippet was truncated**
- **the maximum length of a snippet is 156 characters**
  - As we saw earlier, Google has played with the size
- **Google uses the meta description (if there is one) as the default for a snippet**
- **if there is an Open Directory Project listing for a website, Google uses its meta description over the meta description in the web page**
  - ***http://www.dmoz.org/***
  - The Open Directory Project that uses human editors to organize websites closed as of March, 2017

Google Snippet Lifecycle Changes

USC Viterbi School of Engineering

Classic snippet

Adding images in 2016

Adding videos in 2018

USC

Google Snippet Lifecycle Changes

USC Viterbi School of Engineering

Adding Sitelinks to snippets

Adding Entity Facts

Adding Tables and Lists

https://youtu.be/ezLO7yC4aFo, an 8 minute video Discussing Titles, Snippets and Result Previews

Copyright Ellis Horo

More Facts on Snippets

- **Snippets are computed at query time**
  - They vary depending upon the query
  - the content that ends up in the text snippet can come from anywhere on your page. First sentence, last sentence, footer, call out box
- **If Google determines your site is a discussion forum, in gray text they put out**
  - *"[number] posts – [number] authors – Last post: [some date]"*

- **If Google determines your site is a scholarly article, in gray text they put out**
  - *"by J. Smith – 2010" or "by J. Smith – Cited by 1 – Related articles"*

Snippets Can Vary for a Single Site Depending Upon the Query

USC Viterbi School of Engineering

what cholesterol levels mean

All News Images Shopping Videos More Settings Tools

About 828,000 results (0.58 seconds)

HDL (high-density lipoprotein) **cholesterol** is also called "good" **cholesterol**. HDL protects against heart disease by taking the bad **cholesterol** out of your **blood** and keeping it from building up in your arteries. Your HDL **cholesterol** number is: Low (and considered a risk factor) if it is less than 40.

Understanding Your Cholesterol Numbers | Cleveland Clinic
https://my.clevelandclinic.org/health/.../11920-cholesterol-numbers-what-do-they-mean

About this result Feedback

People also ask

What is a normal cholesterol level?
What is the normal range for cholesterol levels?
What is a healthy cholesterol level UK?
What foods are high in cholesterol?

Feedback

Result for the query "what cholesterol levels mean

A long snippet, and a PAA

Google uses the meta description

cholesterol cleveland clinic

All News Images Shopping Maps More Settings Tools

About 763,000 results (0.51 seconds)

Understanding Your Cholesterol Numbers | Cleveland Clinic
https://my.clevelandclinic.org/health/.../11920-cholesterol-numbers-what-do-they-mea...
When trying to interpret your **cholesterol** numbers, terms like HDL and LDL can be very confusing. ... Your total blood **cholesterol** is a measure of the **cholesterol** components LDL (low-density lipoprotein) **cholesterol**, HDL (high-density lipoprotein) **cholesterol**, and VLDL (very low ...

Cholesterol-Lowering: Heart-Healthy Strategies - Cleveland Clinic
https://my.clevelandclinic.org/.../10842-cholesterol-lowering-heart-healthy-strategies
Cholesterol is a soft, fat-like, waxy substance only found in animal products. Too much **cholesterol** leads to a build-up of fatty materials and debris (called plaque) on the walls of the arteries supplying blood to the heart and other organs. Some **cholesterol** is needed by the body. In fact, **cholesterol** plays a role in normal body ...

A different query "cholesterol Cleveland Clinic" produces the same first result but a different snippet

USC

Copyright Ellis Horowitz 2012 - 2022

Snippets are an Instance of Summarization

- **Automatic summarization** by computer is a traditional subject of *information retrieval*
- Automatic summarization is also part of *machine learning* and *data mining*
- Document summarization tries to create a representative summary or abstract of the entire document, by finding the most informative sentences
- There are two general approaches to automatic summarization:
  - *extraction* *Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary*
  - *abstraction* *abstractive methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might express*
  - *Research to date has focused primarily on extractive methods, which are appropriate for documents, images, and videos*

Copyright Ellis Horowitz 2012-2022

Featured Snippets

- Featured snippets are Google's attempt to answer the query right on the search results page.
  - *Introduced in 2016, Google wants to give the user an immediate answer so they don't have to search the actual results.*
  - *Featured snippets **show up above the #1 ranked spot**, and typically appear above the fold.*
  - *Google pulls snippet answers from pages that rank on Page 1 of the results for that query (spots #1 through #10) - but the page that wins the featured snippet isn't necessarily the #1 result. Google picks the excerpt from the page that best answers the query in a simple, concise format .*

Three Types of *Featured* Snippets

USC Viterbi School of Engineering

1. Paragraph featured snippet

Marketing automation refers to the software that exists with the goal of automating **marketing** actions. Many **marketing** departments have to **automate** repetitive tasks such as emails, social media, and other website actions. The technology of **marketing automation** makes these tasks easier.

What is Marketing Automation? - HubSpot
https://www.hubspot.com/marketing-automation-information

About this result Feedback

2. List featured snippet

14 of the Best College Websites

- University of Maryland. ...
- University of Notre Dame. ...
- Bucknell University. ...
- University of Chicago. ...
- University of Michigan. ...
- Rhode Island School of Design. ...
- George Washington University. ...
- Middlebury College.

More items...

14 of the Best College Websites (And Why They're So Awesome)
https://blog.hubspot.com/marketing/best-college-websites

About this result Feedback

3. Table featured snippet

Google aviation jobs

All News Shopping Maps Videos More Search tools

About 47,600,000 results (0.37 seconds)

Latest 15 Job Listings

Date	ID #	Job Title
7/12/2016	4187	Project Engineer / AOD
7/1/2016	4185	Propulsion Engineer
6/29/2016	4184	A & P Technician
6/29/2016	4183	Accessory Shop Technician

27 more rows, 1 more column

Aviation Jobs | Aviation Job Seekers | Aviation Careers | Aviation ...
jobs.aviationweek.com/ Aviation Week & Space Technology

About this result • Feedback

For a fourth type see: https://www.semrush.com/blog/featured-snippets/

USC

Modifying Your Page to Produce a *Featured* Snippet

USC Viterbi School of Engineering

- Becoming a featured snippet can be achieved by simple on-page adjustments that very clearly define the topic to users
- One of the goals of the featured snippet is to fuel voice search
  - *Create your text so it would answer a query clearly if read back on voice search*?

1. Look for a place in your content to add a "What Is [Keyword]" heading tag.
   - *This sends clear signals to Google that the following text could be used for the featured snippet*
2. Use the "is statement" e.g.
   - *"Agile methodology is a type of project management process, mainly used for software development…"*
3. Define the topic in 2 or 3 sentences
4. Match the featured snippet format: paragraph, bulleted or numbered list, table
5. Don't use first person, e.g. "Our avocados have many health benefits . . . "

- For more details see
- https://searchengineland.com/featured-snippets-the-9-rules-of-optimization-342627

USC

TOO LONG DIDN'T READ (TLDR)
Internet slang

For the query "CRM software", which is a very popular query term, above the fold there are:

- 4 paid ads
- A featured snippet paragraph
- People Also Ask
- A portion of the #1 ranking organic result

- **Conclusion**: it is harder than ever to be found in the organic search results

Above the **fold** refers to a **search engine results page** ranking on the first **page** that is visible without having to scroll down

Copyright E

Extracting a Snippet is Not Always Easy

USC Viterbi School of Engineering

query: "Tesla reports financial results"

one search result and its snippet

portions of the article used to create the snippet; note how long the article is; “financial results” equates to ”earnings”

Ellis

Extracting a Snippet is Not Always Easy Nor Obvious

query: "cloud computing"

one search result and its snippet

"An easy-to-understand introduction"
occurs nowhere in the article
It is in the meta-description
"sit at your PC" occurs lower in the article

Copyright Ellis Horowitz 2012 – 2022

How Does Google Generate Snippets?

One way to find out is to go to patents.google.com and search for all patents with the term "snippets" assigned to Google

Many are patent applications still being reviewed by the patent office

Some are already awarded

Copyright Ellis Horowitz 2012 - 2022

Lets take a closer look
US Patent 8,145,617

USC Viterbi School of Engineering

***Title**:*
Generation of document snippets based on queries and search results

filed: 2005

awarded: 2012

Abstract

A document retrieval system generates snippets of documents for display as part of a user interface screen with search results. The snippet may be generated *based on the type of query or the location of the query terms in the document*. Different snippet generation algorithms may be used depending on the query type. Alternatively, snippets may be generated **based on an analysis of the location of the query terms** in the document

US8145617.pdf - Adobe Acrobat Pro

File Edit View Window Help

Open Create Customize

1 / 11 87.6% Tools Fill & Sign Comment

US008145617B1

(12) **United States Patent**
Verstak et al.

(10) **Patent No.: US 8,145,617 B1**
(45) **Date of Patent: Mar. 27, 2012**

(54) **GENERATION OF DOCUMENT SNIPPETS BASED ON QUERIES AND SEARCH RESULTS**

(75) Inventors: **Alexandre A. Verstak**, Cupertino, CA (US); **Anurag Acharya**, Campbell, CA (US)

(73) Assignee: **Google Inc.**, Mountain View, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **11/282,560**

(22) Filed: **Nov. 18, 2005**

(51) **Int. Cl.**
G06F 7/00 (2006.01)

(52) **U.S. Cl.** **707/708**; 707/711

(58) **Field of Classification Search** 707/708
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,724,571	A	3/1998	Woods	
6,334,132	B1	12/2001	Weeks	
6,374,260	B1 *	4/2002	Hoffert et al.	707/104.1
6,826,553	B1 *	11/2004	DaCosta et al.	707/1
6,915,294	B1 *	7/2005	Singh et al.	707/3
7,032,011	B2 *	4/2006	Woodard et al.	709/220
2005/0131764	A1 *	6/2005	Pearson et al.	705/26

* cited by examiner

Primary Examiner — Fazlul Quader
(74) *Attorney, Agent, or Firm* — Fenwick & West LLP

(57) **ABSTRACT**

A document retrieval system generates snippets of documents for display as part of a user interface screen with search results. The snippet may be generated based on the type of query or the location of the query terms in the document. Different snippet generation algorithms may be used depending on the query type. Alternatively, snippets may be generated based on an analysis of the location of the query terms in the document.

50 Claims, 4 Drawing Sheets

Receive Query
202

Search
204

Copyright Ellis Horowitz 2012 - 2022

Some Guidelines for Snippet Generation

- **Location Based Rules**
  - based on the location of the query terms in page. A paragraph or a portion of a paragraph might be chosen as search results snippets based on the length and distance of the paragraph from the start or end of pages.
  - Every paragraph that includes the query terms is given a score based on the length of the paragraph and the distance of the paragraph from a predetermined location in the document, such as the beginning or the end of the document.
    - documents that include abstracts, executive summaries or comprehensive introductions are identified and used to create a snippet
    - Similarly, the ends of pages can be used if they include a conclusion or summarization
- **Language Dependent Rules**
  - How much of the paragraph are punctuation characters
  - Whether the paragraph ends with punctuation or proposition
  - Whether any of the words in the paragraph is overly long
  - The number of bold or italicized words in the paragraph
- **Rejection rules**
  - Are shorter than a certain threshold
  - Are mostly punctuation, or have punctuation above a certain threshold
  - Contain italicized or bold words above a certain threshold

US Patent 8,145,617 Defines an Algorithm for Snippet Generation

- **The algorithm**
  1. Identify the paragraphs that include the query terms
  2. Score the paragraphs as described below determining the paragraph with the highest score
  3. *Return the phrase in that paragraph that includes the query terms*
- **Quoting from the Detailed Description**
  - The snippet algorithm selects a paragraph that is near the **beginning** of the document if there is an abstract, executive summary, or long introduction. The **end** of the document is used when there is a conclusion or summarization at the end
- **Scoring includes:**
  paragraphs shorter than threshold score 0;
  k-th paragraph from the start gets a score of

kth-positionFactor + max(actual paragraph length, maxParagraphLen)

The paragraph with the highest score is selected for the snippet

Copyright Ellis Horowitz 2012 - 2022

US Patent 8,631,006
Snippets can be based on a User's Profile

USC Viterbi School of Engineering

System and Method for Personalized Snippet Generation
Filed: April 14, 2005
Awarded: Jan. 14, 2014
Abstract:
Snippets of text are generated based in part on a user's profile. An item, such as a document, is examined to identify terms related to the user's profile. A term profile for an identified term is compared to a user's profile. *The more closely related the identified term is to the user's profile, the higher a similarity score will be*. Alternatively, terms found in a document may have a user profile score which may be obtained by looking the term up in the user's profile. *Terms having high profile similarity scores or high user profile scores are used in identifying snippets which may be relevant to a user. The high scoring terms may be added to search terms and provided to a snippet generator*

US008631006B1

(12) **United States Patent**
Haveliwala et al.

(10) **Patent No.: US 8,631,006 B1**
(45) **Date of Patent: Jan. 14, 2014**

(54) **SYSTEM AND METHOD FOR PERSONALIZED SNIPPET GENERATION**

(75) Inventors: **Taher H. Haveliwala**, Mountain View, CA (US); **Sepandar D. Kamvar**, San Francisco, CA (US)

(73) Assignee: **Google Inc.**, Mountain View, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1650 days.

(21) Appl. No.: **11/107,490**

(22) Filed: **Apr. 14, 2005**

(51) **Int. Cl.**
G06F 7/00 (2006.01)
G06F 17/30 (2006.01)

(52) **U.S. Cl.**
USPC **707/732**; 707/722; 707/723

(58) **Field of Classification Search**
USPC 707/3, 5, 722, 723, 732
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,144,944	A	11/2000	Kurtzman, II et al.	
6,275,820	B1	8/2001	Navin-Chandra et al.	
6,701,310	B1 *	3/2004	Sugiura et al.	707/5
7,092,901	B2 *	8/2006	Davis et al.	705/26
7,165,091	B2 *	1/2007	Lunenfeld	709/203
7,418,447	B2 *	8/2008	Caldwell et al.	707/100
2003/0009440	A1 *	1/2003	Inaba et al.	707/1
2004/0034652	A1 *	2/2004	Hofmann et al.	707/102
2004/0236721	A1 *	11/2004	Pollack et al.	707/2
2004/0267723	A1	12/2004	Bharat	
2005/0240580	A1 *	10/2005	Zamir et al.	707/4
2006/0074883	A1 *	4/2006	Teevan et al.	707/3
2006/0112079	A1 *	5/2006	Holt et al.	707/3
2006/0248059	A1 *	11/2006	Chi et al.	707/3

* cited by examiner

Primary Examiner — Apu Mofiz
Assistant Examiner — Jared Bibbee
(74) *Attorney, Agent, or Firm* — Morgan, Lewis & Bockius LLP

(57) **ABSTRACT**

Snippets of text provided are generated based in part on a user's profile. An item, such as a document, is examined to identify terms related to the user's profile. A term profile for an identified term is compared to a user's profile. The more closely related the identified term is to the user's profile, the higher a similarity score will be. Alternatively, terms found in a document may have a user profile score which may be obtained by looking the term up in the user's profile. Terms having high profile similarity scores or high user profile scores are used in identifying snippets which may be relevant to a user. The high scoring terms may be added to search terms and provided to a snippet generator.

13 Claims, 6 Drawing Sheets

Copyright Ellis Horowitz 2012 - 2022

Featured Snippets Results in Google Web Search: An Exploratory Study – Strzelecki, Rutecka

USC Viterbi School of Engineering

Table 1. Type of featured snippet.

featured type	frequency	percentage
paragraph	114465	70,05%
list	46509	28,46%
table	2438	1,49%

Paragraph snippets are the overwhelming type

Table 2. Ranking position for featured snippet

position	frequency	percentage
0	485	0,30%
1	79867	48,87%
2	30618	18,74%
3	20878	12,78%
4	14469	8,85%
5	9582	5,86%
6	2860	1,75%
7	1909	1,17%
8	1319	0,81%
9	860	0,53%
10	554	0,34%

Position 1, the second position on the SERP Is most common

https://www.nwsdigital.com/Blog/What-is-the-Zero-Position#

Table 3. Other snippets displayed along with featured snippet

params	frequency	percentage
image thumbs	102934	62,99%
site links	41348	25,30%
brand	24214	14,82%
wiki	18675	11,43%
ads	3148	1,93%
name	2850	1,74%
map	1807	1,11%
city	1062	0,65%
news	107	0,07%

Google's People Also Ask (PAA) Feature Introduced in 2015 for desktop and mobile

USC Viterbi School of Engineering

how does google generate snippets

All | Videos | News | Images | Shopping | More | Settings | Tools

About 3,830,000 results (0.49 seconds)

Rich Snippets & Star Ratings | Display Reviews & Increase CTR
Ad www.yotpo.com/Rich_Snippet ▾ (646) 655-8389
Get high quality traffic from your paid & organic search results. **Get** a Demo!

Anatomy Of A Google Snippet - Search Engine Land
https://searchengineland.com/anatomy-of-a-google-snippet-38357 ▾
Mar 18, 2010 - This **is a** crucially important detail: **snippets** are determined query-time; in other words, they vary depending on the keyword being searched on, ... price, size, style, manufacturer) would be gathered together — since it would otherwise be unlikely that a **Google-generated snippet** would capture all of this ...

How Google Might Generate Snippets for Search Results
www.seobythesea.com › Search Engine Optimization (SEO) ▾
Feb 25, 2013 - A document retrieval system generates **snippets** of documents for display as part of a user interface screen with search results. The **snippet** may be **generated** based on the type of query or the location of the query terms in the document. Different **snippet** generation algorithms may be used depending on the query type.
You've visited this page 3 times. Last visit: 10/25/17

Google's Featured Snippets: How to Get Your Content to Appear
contentmarketinginstitute.com/2017/03/google-featured-snippets/ ▾
Mar 27, 2017 - The primary thing you need to understand about featured **snippets** is that they **do** not bypass **Google's** complex ranking system. They are among the topic organic results for a query. Because of this, it's critical to keep all the standard ranking requirements in mind as you create featured content. This means ...

People also ask
- How does Google select featured snippets?
- What is a snippet in Google?
- What is the use of snippets SEO?
- What is snippet generation?

Feedback

PAA result for query "how does google generate snippets

People also ask

How does Google select featured snippets?

Here are a few simple steps I've used to create content that ranks in the snippets.

1. Create content specifically to answer questions. Provide in-depth answers. ...
2. Know the questions your readers are asking. ...
3. Create truly high-quality content. ...
4. Work to provide the best answer. ...
5. Use question-and-answer pages.

Google's Featured Snippets: How to Get Your Content to Appear
contentmarketinginstitute.com/2017/03/google-featured-snippets/

Search for: How does Google select featured snippets?

What is a snippet in Google?

Rich Snippets is the term used to describe structured data markup that site operators can add to their existing HTML, which in turn allow search engines to better understand what information is contained on each web page.

A Beginner's Guide to Rich Snippets | Unamo Blog
https://unamo.com/blog/seo/beginners-guide-rich-snippets

Search for: What is a snippet in Google?

- What is the use of snippets SEO?
- What is snippet generation?
- What is a featured snippet?
- What is a snippet of a song?
- What is the code snippet?

Feedback

Expansion of People Also Ask

In one study, the "People Also Ask" box appeared on 364 keywords out of 1,788, 20%.

USC

Copyright Ellis Horowitz 2012 - 2022

People Also Ask (PAA) is Growing Fast

- The "People Also Ask" box is a Google universal SERP result that answers questions related to the searcher's initial query.
- It is a cousin of the featured snippet
- Each PAA box contains anywhere from one to four related questions which expand to reveal answers that Google has pulled from other websites
- The site's URL appears below each answer, along with a "Search for" link, which guides the user to a Google SERP of the PAA question.

Use of PAAs are growing faster than snippets according to https://moz.com/blog/infinite-people-also-ask-boxes

Copyright Ellis Horowitz 2012 - 2022

Rich Snippets

- **In 2009**, Google announced *Rich Snippets*, a mechanism ***for website developers*** to include information that Google's results algorithm will *display as a snippet*
- The mechanism calls for *embedding structured data in web pages* with the objective of displaying the structured data to a user in a visually outstanding way.
- Rich Snippets give users a convenient summary information about their search results at a glance.

For example, the results for Club Deluxe includes internal data such as:

address

hours

directions

Rich Snippets Examples: People Snippets

USC Viterbi School of Engineering

Google

pravir gupta | Search

About 13,400 results (0.28 seconds) Advanced search

Everything
More

Show search tools

Pravir Gupta | Facebook
Friends: Sam Tyagi, Geeta Shroff, Siddarth Jain, Shradha Balakrishnan, Richa Kumar
Pravir Gupta is on Facebook. Join Facebook to connect with **Pravir Gupta** and others you may know. Facebook gives people the power to share and makes the ...
www.facebook.com/**pravirgupta** - Cached

Home (**pravir**)
Pravir Gupta. ... attachment removed by **Pravir Gupta**. edited by **Pravir Gupta** ... created by **Pravir Gupta**. Home. created by **Pravir Gupta** ...
pravirgupta.com/ - Cached - Similar

Pravir Gupta - Knol: a unit of knowledge
Pravir Gupta. Verify Name. Agra, India. Public activity feed. Sort by: ... by**Pravir Gupta**. We are continuously looking at enabling sites. ...
knol.google.com/k/**pravir-gupta**/-/3phlmrwubhfji/0 - Cached - Similar

The Journey is the Reward - a knol by **Pravir Gupta**
Jul 20, 2009 ... Debut novel by Anil Kumar **Gupta** which was published in July 2009.
knol.google.com/k/**pravir-gupta**/the-journey-is-the-reward/.../4 - Cached

Show more results from knol.google.com

Pravir Gupta - Senior Software Engineer | LinkedIn
San Francisco Bay Area - Senior Software Engineer
View **Pravir Gupta's** (87 connections) professional profile on LinkedIn. LinkedIn is the world's largest business network, helping professionals like Pravir ...
www.linkedin.com/pub/**pravir-gupta**/2/180/a70 - Cached - Similar

Pravir Gupta - Directory | LinkedIn
View the profiles of professionals named **Pravir Gupta** on LinkedIn. There are 2 professionals named **Pravir Gupta** who use LinkedIn to exchange information, ...
www.linkedin.com/pub/dir/**Pravir**/**Gupta**/

Pravir Gupta, Google Inc, Mountain View, CA | Spoke
Pravir Gupta, Google Inc of Google Inc"s information - including email, business address, business phone, biography, title, company, jobs and associations, ...
www.spoke.com/info/p90ookh/**PravirGupta** - Cached

here the snippets describe the pages containing the information about the individual: Facebook, LinkedIn, Google

USC

Rich Snippets Examples: Events

Google

fillmore events | Search

About 1,610,000 results (0.21 seconds) Advanced search

Everything
More

All results
Timeline
More search tools

The **Fillmore** Concert Tickets, Schedule, Seating Chart | Official ...
Get email alerts and never miss your favorite **events** at The **Fillmore**. Please enter your e-mail address. That is not a valid e-mail address format. ...
www.the**fillmore**.com/ - Cached - Similar

The **Fillmore** San Francisco - The **Fillmore** Schedule | Eventful
View The **Fillmore's** upcoming **event** schedule and profile - San Francisco, CA. The **Fillmore**, also known as **Fillmore** Auditorium, is located in San ...

Carolina Chocolate Drops	Thu, Jun 24
Josh Ritter & the Royal City Band	Thu, Jun 24
Robert Earl Keen	Sat, Jun 26

eventful.com › San Francisco venues - Cached - Similar

Fillmore Events: Events in Fillmore, California
Fillmore Events Directory. Includes listings for Events in Fillmore, California.
www.californiacoast-worldweb.com/**Fillmore**/**Events**/ - Cached - Similar

San Francisco The **Fillmore Events**, Shows & Things to do - SF Gate
Find 48 San Francisco The **Fillmore events** and show tickets and more on Zvents. Popular The **Fillmore Events** are Salsa Festival on the Fillmore, Fillmore Jazz ...
events.sfgate.com/san-francisco-ca/**events**/the+**fillmore** - Cached

New York **Fillmore Events** Events, Shows & Things to do - NY Daily News
Find 29 New York **Fillmore Events** events and show tickets and more on Zvents. Popular **Fillmore Events** Events are On Fillmore Plus Rachel Grimes, ...
events.nydailynews.com/new-york-ny/**events**/**fillmore+events** - Cached

Charlotte Charlotte **Fillmore Events**, Shows & Things to do - The ...
Find 8 Charlotte Charlotte **Fillmore events** and show tickets and more on Zvents. Popular Charlotte **Fillmore Events** are Smashing Pumpkins, Adam Lambert with ...
events.charlotteobserver.com/charlotte-nc/**events**/charlotte+**fillmore** - Cached

the Filmore theatre can highlight future concerts by regularly updating their webpage with the latest rich snippet information

Advantages of *Rich* Snippets

USC Viterbi School of Engineering

Benefits of Rich Snippets in Google Search …

- **Webmasters:** Provides webmasters the ability to add useful information to their web search result snippets to help Google make sense of their bits.
- **Purpose** Provides more information to a user about the content that exists on page so they can decide which result is more relevant for their query.
- Two good reasons for using rich snippets

1. **Additional traffic to a webpage** With extra information people tend to rely more on a particular search result with linked data, thus an increasing number of impressions noted on sites with Rich Snippets.
2. **Higher Click Through Rate** An increasing number of higher click-through rate for pages with Rich Snippets was experienced as shown in a paper by *Kavi Goel, Pravir Gupta*
   - *http://www.dataversity.net/google-yahoo-and-bing-announce-schema-org/*

- **Easy to add** simple lines of Markup to existing HTML, no affect to visual appearance of the webpage.

A Joint Effort by Google, Yahoo! And Bing

USC Viterbi School of Engineering

- In June, 2011 Google, Yahoo, and Bing agree on a single standard
- They establish the website schema.org which defines the mechanism for creating rich snippets
- They decide to standardize on microdata format
- https://developers.google.com/structured-data/rich-snippets/

Google, Yahoo! and Bing Announce Schema.org

By Eric Franzon / June 2, 2011 / 0 Comments

0 0

[Revised and re-posted at 4:03pm EST]

schema.org

In a collaborative effort reminiscent of sitemaps.org, Google, Yahoo! and Bing have announced the launch of schema.org. Perhaps the most significant aspect of this announcement is the particular standard they have focused on: namely, microdata.

In the Google announcement, Kavi Goel and Pravir Gupta of Google's search team say, "Historically, we've supported three different standards for structured data markup: microdata, microformats, and RDFa. We've decided to focus on just one format for schema.org to create a simpler story for webmasters and to improve consistency across search engines relying on the data."

From the Yahoo! announcement comes this: "Today's announcement offers tremendous opportunity for growth. In addition to consolidating the schemas for the vocabularies we already support, there are schemas for more than a hundred newly created categories including movies, music, organizations, TV shows, products, places and more. We will continue to expand these categories by listening to feedback from the community and will continue publishing new schemas on a regular basis. Don't worry if your site has already added RDFa or microformats currently supported by our Enhanced Displays program, that site will still appear with an Enhanced Display on Yahoo! – no changes required."

And Bing has this to add: "At Bing we understand the significant investment required to implement markup, and feel strongly that by partnering with Google and Yahoo! on standard schemas webmasters can be more efficient with the time they invest… Bing accepts a wide variety of markup formats today (Open Graph, microformat, etc.) for features like Tiles and will continue to do so, but by standardizing on schema.org we are looking to simplify the markup choices for webmasters and amplify the value the receive in return."

The schema.org site "provides a collection of schemas, i.e., html tags, that webmasters can use to markup their pages in ways recognized by major search providers. Search engines…rely on this markup to improve the display of search results, making it easier for people to find the right web pages."

Rich Snippet Technology Definitions

USC Viterbi School of Engineering

- Google suggests using the microdata formalism for snippets http://www.whatwg.org/specs/web-apps/current-work/multipage/microdata.html

https://www.w3.org/TR/microdata/
Now goes to
https://html.spec.whatwg.org/multipage/

- Two other formalisms for creating rich snippets have been suggested:
- RDFa (Resource Description Framework – in Attributes) http://en.wikipedia.org/wiki/RDFa
- Microformat Encoding http://en.wikipedia.org/wiki/Microformat

USC

Schema.org Vocabulary

- Schema.org defines an object hierarchy
- The most general item type is Thing with properties: name, description, url, and image
  - Person, Place and Organization are types of Things
  - More specific items inherit the properties of their parent
- Some commonly used types include:
  - Creative works: book, movie, music recording, recipe, TV Series
  - Embedded object: image, video
  - Event
  - Organization
  - Person
  - Place, LocalBusiness, e.g. Restaurant
  - Product, Offer, Aggregate Offer
  - Review, AggregateRating

Entities in *Rich* Snippet Encodings

USC Viterbi School of Engineering

Entities supported by Google Rich Snippets as of now….

- Software applications
- Breadcrumbs
  - a breadcrumb trail on a page indicates the page's position in the site hierarchy. A user can navigate all the way up in the site hierarchy, one level at a time, by starting from the last breadcrumb in the breadcrumb trail
  - for example, Books › Authors › Ann Leckie › Ancillary Justice
- Events
- Music
- Businesses and Organizations
- People
- Products
- Recipes
- Review Ratings
- Reviews: should include: item being reviewed, reviewer rating, date
- Videos: Facebook Share

Rich Snippets

- ***Microformats*** use only existing HTML, e.g. the ***class*** attribute in HTML tags (often or) to assign brief and descriptive names to entities and their properties
- ***Microdata*** extends HTML5 by introducing new attributes like itemprop
- **Microformat Example**

```
<div class="vcard">
    <img class="photo" src="www.example.com/bobsmith.jpg" />
    <strong class="fn">Bob Smith</strong>
    <span class="title">Senior editor</span> at <span class="org">ACME Reviews</span>
    <span class="adr">
        <span class="street-address">200 Main St</span>
        <span class="locality">Desertville</span>, <span class="region">AZ</span>
        <span class="postal-code">12345</span>
    </span>
</div>
```

microformat class attributes in this example include vcard, photo, title, org, adr, locality, etc

A MicroData Example: A Web Page About the Movie Avatar

- **To begin, identify the section of the page that is "about" the movie Avatar. To do this, add the itemscope element to the HTML tag that encloses information about the item, and you can specify the type of item using the itemtype attribute like this:**

```
<div itemscope itemtype="http://schema.org/Movie">
<h1>Avatar</h1>
<span>Director: James Cameron (born August 16, 1954) </span>
 <span>Science fiction</span>
<a href="../movies/avatar-theatrical-trailer.html">Trailer</a>
</div>
```

- **By adding itemscope, you are specifying that the HTML contained in the … block is about a particular item.**

Avatar Example Continued

- **The itemprop attribute is used to label properties of a movie such as actors, director, ratings.**
- **For example, to identify the director of a movie, add itemprop="director" to the element enclosing the director's name. (There's a full list of all the properties you can associate with a movie at http://schema.org/Movie.)**

```
<div itemscope itemtype ="http://schema.org/Movie">
<h1 itemprop="name">Avatar</h1>
<span>Director: <span itemprop="director">James Cameron</span> (born
    August 16, 1954)</span>
<span itemprop="genre">Science fiction</span>
<a href="../movies/avatar-theatrical-trailer.html" itemprop="trailer">Trailer</a>
</div>
```

Partial List of Movie Properties (Schema.org/Movie)

USC Viterbi School of Engineering

Movie - schema.org - Mozilla Firefox

File Edit View History Bookmarks Tools Help

schema.org/Movie

Movie - schema.org

Thing > CreativeWork > Movie

A movie.

Property	Expected Type	Description
Properties from Thing		
description	Text	A short description of the item.
image	URL	URL of an image of the item.
name	Text	The name of the item.
url	URL	URL of the item.
Properties from CreativeWork		
about	Thing	The subject matter of the content.
accountablePerson	Person	Specifies the Person that is legally accountable for the CreativeWork.
aggregateRating	AggregateRating	The overall rating, based on a collection of reviews or ratings, of the item.
alternativeHeadline	Text	A secondary title of the CreativeWork.
associatedMedia	MediaObject	The media objects that encode this creative work. This property is a synonym for encodings.
audio	AudioObject	An embedded audio object.
author	Person or Organization	The author of this content. Please note that author is special in that HTML 5 provides a special mechanism for indicating authorship via the rel tag. That is equivalent to this and may be used interchangeably.
award	Text	An award won by this person or for this creative work.
awards	Text	Awards won by this person or for this creative work. (legacy spelling; see singular form, award)
comment	UserComments	Comments, typically from users, on this CreativeWork.
contentLocation	Place	The location of the content.
contentRating	Text	Official rating of a piece of content—for example,'MPAA PG-13'.
contributor	Person or Organization	A secondary contributor to the CreativeWork.
copyrightHolder	Person or Organization	The party holding the legal copyright to the CreativeWork.

Movie - schema.org - Mozilla Firefox

File Edit View History Bookmarks Tools Help

schema.org/Movie

Movie - schema.org

copyrightYear	Number	The year during which the claimed copyright for the CreativeWork was first asserted.
creator	Person or Organization	The creator/author of this CreativeWork or UserComments. This is the same as the Author property for CreativeWork.
dateCreated	Date	The date on which the CreativeWork was created.
dateModified	Date	The date on which the CreativeWork was most recently modified.
datePublished	Date	Date of first broadcast/publication.
discussionUrl	URL	A link to the page containing the comments of the CreativeWork.
editor	Person	Specifies the Person who edited the CreativeWork.
encoding	MediaObject	A media object that encode this CreativeWork.
encodings	MediaObject	The media objects that encode this creative work (legacy spelling; see singular form, encoding).
genre	Text	Genre of the creative work
headline	Text	Headline of the article
inLanguage	Text	The language of the content. please use one of the language codes from the IETF BCP 47 standard.
interactionCount	Text	A count of a specific user interactions with this item—for example, 20 UserLikes, 5 UserComments, or 300 UserDownloads. The user interaction type should be one of the sub types of UserInteraction.
isFamilyFriendly	Boolean	Indicates whether this content is family friendly.
keywords	Text	The keywords/tags used to describe this content.
mentions	Thing	Indicates that the CreativeWork contains a reference to, but is not necessarily about a concept.
offers	Offer	An offer to sell this item—for example, an offer to sell a product, the DVD of a movie, or tickets to an event.
provider	Person or Organization	Specifies the Person or Organization that distributed the CreativeWork.
publisher	Organization	The publisher of the creative work.
publishingPrinciples	URL	Link to page describing the editorial principles of the organization primarily responsible for the creation of the

USC

Copyright Ellis Horowitz 2012 - 2022

MicroData Markup for "Pirates of the Caribbean"

USC Viterbi School of Engineering

```
<div itemscope itemtype="http://schema.org/Movie">
<h1 itemprop="name">Pirates of the Carribean: On Stranger Tides (2011)</h1>
<span itemprop="description">Jack Sparrow and Barbossa embark on a quest to find the elusive fountain of
    youth, only to discover that Blackbeard and his daughter are after it too.</span>
Director: <div itemprop="director" itemscope itemtype="http://schema.org/Person">
        <span itemprop="name">Rob Marshall</span> </div>
Writers:
 <div itemprop="author" itemscope itemtype="http://schema.org/Person">
         <span itemprop="name">Ted Elliott</span>  </div>
<div itemprop="author" itemscope itemtype="http://schema.org/Person">
        <span itemprop="name">Terry Rossio</span>  </div>   , and 7 more credits
Stars:
 <div itemprop="actor" itemscope itemtype="http://schema.org/Person">
        <span itemprop="name">Johnny Depp</span>,   </div>
<div itemprop="actor" itemscope itemtype="http://schema.org/Person">
        <span itemprop="name">Penelope Cruz</span>,  </div>
<div itemprop="actor" itemscope itemtype="http://schema.org/Person">
       <span itemprop="name">Ian McShane</span>   </div>
<div itemprop="aggregateRating" itemscope itemtype="http://schema.org/AggregateRating">
     <span itemprop="ratingValue">8</span>/<span itemprop="bestRating">10</span> stars from
     <span itemprop="ratingCount">200</span> users.
 Reviews: <span itemprop="reviewCount">50</span>. </div> </div>
```

Includes

Movie name
Description
Director
Author
Actors
rating

USC

More Examples: Clarifying Hard to Understand Content

USC Viterbi School of Engineering

- **The <time> element has attributes: dates, times and durations:**
  - <time datetime="2022-04-01">04/01/22</time>
  - <time datetime=2022-05-08T19:30">May 8, 7:30pm</time>
  - <time itemprop="cookTime" datetime=PT1H30M">1 ½ hrs</time>
- **Here is markup for a concert on May 8, 2022**<div itemscope itemtype="http://schema.org/Event">

```
<div itemprop="name">Spinal Tap</div>
<span itemprop="description">One of the loudest bands ever reunites for an unforgettable
    two-day show.</span>
Event date:
<time itemprop="startDate"
            datetime="2022-05-08T19:30">May 8, 7:30pm</time>
</div>
```

- **Here is markup for an enumeration**

```
<div itemscope itemtype="http://schema.org/Offer">
  <span itemprop="name">Blend-O-Matic</span>
  <span itemprop="price">$19.95</span>
  <span itemprop="availability">Available today!</span> </div>
```

USC

https://technicalseo.com/tools/schema-markup-generator/

USC Viterbi School of Engineering

a web interface tool for creating a rich snippet

Copyright Ellis Horowitz 2012 - 2022

USC

https://search.google.com/test/rich-results

USC Viterbi School of Engineering

Results for www.cs.usc.edu

Results for www.usc.edu

Copyright Ellis Horowitz 2012 - 2022

Google's Structured Testing Tool

https://search.google.com/structured-data/testing-tool

Google has created a tool for examining a web site with microformat data and indicating if there are any errors, e.g. http://www.urbanspoon.com/r/6/765421/restaurant/Pizza-My-Heart-santa-cruz

Copyright Ellis Horowitz 2012 - 2022

Google's *Rich* Snippets Testing Tool

https://developers.google.com/structured-data/testing-tool/

USC Viterbi School of Engineering

Structured Data Testing To

https://developers.google.com/structured-data/testing-tool/

Apps | Bookmarks | CSCI 571 Home Page | CSCI 572 Home Page | Computer Science De | ITS - Software | Horowitz Family | CS276: Information Re | NLP | Coursera IR Videos | Other bookmarks

Google Developers | Structured Data Testing T... | Search | ehorowitz1@gmail.com Change account | Sign out

Fetch URL | Examples | VALIDATE

```
1  <!-- Add this code to person page: -->
2      <div itemscope itemprop="author"
   itemtype="http://schema.org/Person">
3      <meta itemprop="description" content="Dr. Ellis Horowitz is
   currently Professor of Computer Science and Electrical Engineering
4      at the University of Southern California.
5      The company designed and developed UNIX application
   software.">
6          <span itemprop="name" style="display:block;"><strong>Ellis
   Horowitz</strong></span>
7          <span itemprop="member" itemscope
   itemtype="http://schema.org/Organization"
   style="display:block;">University of Southern California</span>
8          <span itemprop="jobtitle"
   style="display:block;">Professor</span>
9          <span itemprop="email"
   style="display:block;">ehorowitz1@gmail.com</span>
10         <div itemprop="address" itemscope
   itemtype="http://schema.org/PostalAddress"><span
   itemprop="postalCode"style="display:block;">90089</span>
11             </div>
```

Results - Filter by use case

Person (1) — 1 Error

Person	
description:	Dr. Ellis Horowitz is currently Professor of Computer Science and Electrical Engineering at the University of Southern California. The company designed and developed UNIX application software.
name:	Ellis Horowitz
email:	ehorowitz1@gmail.com
jobTitle:	Professor
member [Organization]:	
address [PostalAddress]:	
postalCode:	90089

Custom Search Result Filters

Connect | Programs | Developer Consoles | Explore

USC

Google Introduces New Tags for Snippet Control(1)

- The robots meta tag is added to an HTML page's <head>; here are some new tags:
  - `"nosnippet"`
    This is an existing option to specify that you don't want any textual snippet shown for this page.
  - `"max-snippet:[number]"`
    New! Specify a maximum text-length, in characters, of a snippet for your page.
  - `"max-video-preview:[number]"`
    New! Specify a maximum duration in seconds of an animated video preview.
  - `"max-image-preview:[setting]"`
    New! Specify a maximum size of image preview to be shown for images on this page, using either "none", "standard", or "large".
- They can be combined, for example:

```
<meta name="robots" content="max-snippet:50, max-image-preview:large">
```

Google Introduces New Tags for Snippet Control(2)

- A new way to help limit which part of a page is eligible to be shown as a snippet is the `"data-nosnippet"` HTML attribute on span, div, and section elements.
  - With this, you can prevent that part of an HTML page from being shown within the textual snippet on the page.
- For example:
- `<p><span data-nosnippet>Harry Houdini</span> is undoubtedly the most famous magician ever to live.</p>`
- **To opt out of featured snippets**
- The nosnippet tag blocks all snippets (featured snippets and regular snippets) for the tagged page.
- Text marked by the data-nosnippet tag won't appear in featured snippets (or regular snippets either).
- If both nosnippet and data-nosnippet appear in a page, nosnippet takes priority, and snippets won't be shown for the page.

Summary

- **Snippets can be divided into five categories**
  1. ***Regular snippets,*** *displayed in organic search results*
  2. ***Rich snippets*** *come from structured data dictionary schema.org including RDFa, Microdata or JSON*
  3. ***Google News,*** *created automatically from news feeds to Google*
  4. ***Entity types,*** *come from the KnowledgeGraph, are constructed object and concepts including people, movies, places, events, books, etc*
  5. ***Features snippets,*** *determine that a page contains a likely answer to the user's question; the snippet is displayed. In four different forms: paragraph, table, ordered list, unordered list*

Extending Schema.org to handle PAA

USC Viterbi School of Engineering

- **QAPage focuses on a specific question and its answer(s)**
- **https://schema.org/QAPage**
- **Question, a specific question from a user seeking answers online or collected in a FAQ document**
- **https://schema.org/QAPage**
- **HowTo, instructions that explain how to achieve a results by performing a sequence of steps**
- **https://schema.org/HowTo**
- **Here is an article on infinite PAAs, https://moz.com/blog/infinite-people-also-ask-boxes**
- **Matt Cutts Discusses Snippets**
  - **https://www.youtube.com/watch?v=vS1Mw1Adrk0**
  - **https://www.youtube.com/watch?v=NlJiLDn9-38**
- Three useful webpages on rich snippets:

http://support.google.com/webmasters/bin/answer.py?hl=en&answer=99170&topic=21997&ctx=topic

http://support.google.com/webmasters/bin/answer.py?hl=en&answer=1093493

USC

Some Interesting Videos

- Web Duplication
- https://www.youtube.com/watch?v=LsfIlviyDvE

- Rendering
- https://www.youtube.com/watch?v=rq8sFkl0KnI

- Googlebot and Web Hosting (9 min)
- https://www.youtube.com/watch?v=JvYh1oe5Zx0

- Titles, Snippets and Result Previews
- https://youtu.be/ezLO7yC4aFo

Clustering

Copyright Ellis Horowitz 2011-2022

Today's Topic: Clustering

- **Document clustering**
  - Motivations
  - Document representations
  - Success criteria
- **Clustering algorithms**
  - Partitional
  - Hierarchical

Copyright Ellis Horowitz, 2011-2022

What is Clustering?

- **Clustering: the process of grouping a set of objects into classes of similar objects**
  - Documents within a cluster should be similar.
  - Documents from different clusters should be dissimilar.
- **Clustering is the most common form of *unsupervised learning***
  - Unsupervised learning = learning from raw data, as opposed to supervised learning where a classification of examples is given a priori
- Clustering is a common and important task that finds many applications in IR and other places

Copyright Ellis Horowitz, 2011-2022

What is NOT Clustering

- Supervised classification
  - Have class label information
- Simple segmentation
  - Dividing students into different registration groups alphabetically, by last name
- Results of a query
  - Groupings are a result of an external specification
- Graph partitioning
  - Some mutual relevance and synergy, but areas are not identical

Copyright Ellis Horowitz, 2011-2022

Applications of Clustering 3 Scenarios

1. A telephone company needs to establish its network by putting its towers in a particular region it has acquired. The location of putting these towers can be found by using a clustering algorithm so that all its users receive optimum signal strength

2. The Miami DEA wants to make its law enforcement more stringent and hence have decided to make their patrol vans stationed across the area so that the areas of high crime rates are in the vicinity to the patrol vans

3. A hospital care chain wants to open a series of Emergency-Care wards, keeping in mind the factor of maximum accident prone areas in a region

Copyright Ellis Horowitz, 2011-2022

Why Search Engines Cluster Documents

1. **For improving recall in search applications**
   - Better search results; similar documents are grouped
2. **For speeding up vector space retrieval**
   - Faster search if clustering occurs a priori
3. **Cleaner user interface**
4. **Automatic thesaurus generation by clustering related terms**

- ***Cluster hypothesis* - Documents with similar text are related**
- **Ergo, to improve search recall:**
  - In theory we could cluster docs in our corpus a priori so

  when a query matches a doc *D*, we also return other docs in the cluster containing *D*
  - ***This strategy doesn't work for search engines***

Copyright Ellis Horowitz, 2011-2022

Related Searches are a Form of Clustering

Google related searches

Yahoo does some clustering via alternate queries

Bing does a little better

Copyright Ellis Horowitz, 2011-2022

yippy.com Search Engine

- **Yippy** (formerly **Clusty**) is a metasearch engine developed by Vivísimo which emphasizes clusters of results.

initial screen
with query "cars"

clustered results appear
on the left column: e.g.
sale
reviews
dealers
rentals

multiple level clusters:
car dealers
trucks
ebay

Copyright Ellis Horowitz, 2011-2022

Yahoo's Name Derives from *Y*et *A*nother *H*ierarchical *O*fficious *O*racle

Yahoo! Hierarchy *isn't* clustering but *is* the kind of output you want from clustering – a taxonomy

See

https://searchengineland.com/yahoo-directory-close-204370

Copyright Ellis Horowitz 2011-2022

Google News: Automatic Clustering Gives an Effective News Presentation Metaphor

recent Supreme court decisions clustered together

Typical newspaper clusters: World, US, Business, Technology, Sports, etc

These clusters must be constantly re-computed to make sure the latest news is included

Clustering Examples from Google RSS Feeds

Two examples of Google feeds
There is a main article, some text and beneath that related or clustered articles

Tesla's Model 3 Market Opportunity Is Bigger Than You Think
Motley Fool
Tesla's (NASDAQ:TSLA) forthcoming Model 3 will be unveiled next month, go into production in late 2017, cost about $35,000 before incentives, and ...

Tesla Signs Lease for 40K-SF Red Hook Dealership - Commercial Observer
Advertising enters the equation for **Tesla** Motors - Seeking Alpha
Tesla Motors Finally Gets Its Paws on **Tesla**.com - Inverse
Full Coverage

There's one new **Tesla** car that nobody is talking about
Businessinsider India
These two **Teslas** are all anyone has been talking about lately - especially Wall Street analysts who want to figure out which way **Tesla's** extremely ...

VIDEO: **Tesla** Drag Race! Model S vs. Model X In STUNNING Showdown - AutoSpies.com
Tesla Model S & Model X Comparison (Price, Range, Acceleration) After Removal Of 85 kWh Version - InsideEVs
Tesla Will Begin Taking Preorders on Its Make-or-Break Vehicle - GreatNews
Full Coverage

Copyright Ellis Horowitz 2011-2022

What Is A Good Clustering?

- **Internal criterion: A good clustering will produce high quality clusters in which:**
  - the **intra-class** (that is, intra-cluster) similarity is high
  - the **inter-class** similarity is low
  - The measured quality of a clustering depends on both the document representation and the similarity measure used

Copyright Ellis Horowitz, 2011-2022

Three Criteria of Adequacy for Clustering Methods

1. The method produces a clustering which is **unlikely to be altered drastically** when further objects are incorporated
   – i.e. it is stable even under significant growth
2. The method is **stable** in the sense that small errors in the description of objects lead to small changes in the clustering
3. The method is **independent** of the initial ordering of the objects

Copyright Ellis Horowitz, 2011-2022

Classification is Different from Clustering

- In general, in ***classification*** you have a set of predefined classes and want to know which class a new object belongs to.
- ***Clustering*** tries to group a set of objects and find whether there is *some* relationship between the objects.
  - Clustering *precedes* classification
- In the context of machine learning, classification is *supervised learning* and clustering is *unsupervised learning*
  - **Clustering** requires a. an algorithm, b. a similarity measure, and c. a number of clusters
  - **classification** has each document labeled in a class and an algorithm that assigns new documents to one of the classes

Copyright Ellis Horowitz, 2011-2022

Begin with Clustering

- **Step 1: Given a large set of computer science documents, first we cluster them using some algorithm (to be presented)**

Copyright Ellis Horowitz, 2011-2022

Then We Name the Clusters

- **Step 2: we label the clusters**
  - **choosing a popular name from each document cluster**

Copyright Ellis Horowitz, 2011-2022

Still Clustering: Determine Decision Boundaries

- **Step 3: we compute boundaries for the clusters that can be used as new documents appear; i.e. classification**

Copyright Ellis Horowitz, 2011-2022

Classification Requires Initial Clusters and Boundaries

- **Definition:** *Supervised Learning*, inferring a function from labeled training data

1. **The documents in each cluster define the "training" docs for each category**
   - E.g in computer science named clusters would include: Algorithms, Theory, AI, Databases, Operating Systems, NLP, etc.
2. **Documents are in a cluster based upon the similarity measure used;**
   - generally a vector space with each doc viewed as a bag of words
3. **A classifier is an algorithm that will classify new docs**
   - Essentially, the decision space is partitioned and an algorithm is devised
4. **Given a new doc, the new algorithm determines which partition it falls into**

Copyright Ellis Horowitz, 2011-2022

Now Let's Return to the Earlier Problem: Clustering

- **Questions to consider when clustering**
  - How do we represent the document?
    - *Usually as a vector space*
  - How do we compute similarity/distance?
    - *Using cosine similarity*
  - How many clusters?
    - *will it be a fixed a priori number? or*
    - *completely data driven?*
  - Be careful to avoid "trivial" clusters - too large or small
    - *If a cluster is too large, then for navigation purposes you've wasted an extra user click without whittling down the set of documents much*

Copyright Ellis Horowitz, 2011-2022

Issue: Hard vs. Soft Clustering

- ***Hard clustering*: Each document belongs to exactly one cluster**
  - More common and easier to do
- ***Soft clustering*: A document can belong to more than one cluster.**
  - Makes sense for some applications e.g. news about Los Angeles might be included in local and national news clusters
  - E.g. you may want to put a pair of sneakers in two clusters: (i) sports apparel and (ii) shoes

Copyright Ellis Horowitz, 2011-2022

What Definition of Similarity/Distance Will Be Used

- **Once again we will treat documents as vectors**
  - **Cosine similarity (seen before many times)**
    - Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. Range from 0 (dissimilar) to 1 (exactly similar)

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

  - **Most clustering implementations use cosine similarity**
  - **Euclidean distance is a close alternative that is also popular**

Copyright Ellis Horowitz, 2011-2022

A Data Set with Clear Cluster Structure

Circles represent documents as N-vectors

- **How would you design an algorithm for finding the three clusters in this case?**
- **Hint: use a distance measure**

Clustering Algorithms

- **Two general methodologies**
  - Partitioning Based Algorithms
  - Hierarchical Algorithms
- **Partitioning Based**
  - Choose K and then divide a set of N items into K clusters
- **Hierarchical – Bottom Up/Top Down**
  - agglomerative: pairs of items or clusters are successively linked to produce larger clusters (hierarchy produced bottom-up)
  - divisive: start with the whole set as a cluster and successively divide sets into smaller partitions (hierarchy produced top-down)

Copyright Ellis Horowitz, 2011-2022

A Partitioning Algorithm
K-Means Clustering Algorithm

- **Clustering algorithm strategy**
  - Choose *k* random data items out of the *n* items; call these items the *means*; they designate the prototype or name of the cluster
  - **Refine it iteratively**
    - Associate each of the *n-k* items with one of the ***k* clusters** choosing the **cluster** that it is nearest to**;**
    - **This is called *K*-means clustering**
- **Recall**
  - The "***mean***" is the "average" where you add up all the numbers and then divide by the number of numbers.
  - The "***median***" is the "middle" value in the list of numbers. To find the median, you may have to sort
  - The "***mode***" is the value that occurs most often. If no number is repeated, then there is no mode for the list

Copyright Ellis Horowitz, 2011-2022

Different Ways of Clustering the Same Set of Points

K-means clustering critically depends upon the value of k

Copyright Ellis Horowitz 2011-2022

"Optimal" K-Means Clustering

- The **optimal *k*-means clustering problem** calls for finding cluster centers that minimize the intra-class variance, i.e. the sum of squared distances from each data point being clustered to its cluster center;
- **The problem stated formally**:
  - Given a finite set S where each element is a vector of length d, find a subset T of size k that minimizes the sum of squares of the distances between elements in S and their closest element in T
- Finding an exact solution to the *k*-means problem for arbitrary input has been shown to be **NP-hard**
- **NP**-hardness (non-deterministic polynomial-time **hard**), in computational complexity theory, is a class of problems that are, informally, "at least as **hard** as the hardest problems in **NP**".
- finding a polynomial algorithm to solve any NP-hard problem would give polynomial algorithms for all the problems in NP, which is unlikely

Copyright Ellis Horowitz, 2011-2022

K-Means Clustering Algorithm Mathematical Formulation

(***stated mathematically***)

Given an initial set of k means $m_1^{(1)},\ldots,m_k^{(1)}$, the algorithm proceeds by alternating between two steps:

Assignment step: Assign each observation to the cluster whose mean yields the least within-cluster sum of squares. Since the sum of squares is the squared Euclidean distance, this is intuitively the "nearest" mean

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \ \forall j, 1 \leq j \leq k\},$$

where each x_p is assigned to exactly one $S^{(t)}$, even if it could be assigned to two or more of them.

Update step: Calculate the new means to be the centroids of the observations in the new clusters.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

- The algorithm has converged when the assignments no longer change.
- The algorithm will converge to a (local) optimum.
- There is no guarantee that the global optimum is found using this algorithm.

Copyright Ellis Horowitz 2011-2022

An Approximation Clustering Algorithm

1. **Select K points as initial centroids**
2. **repeat**
   - form K clusters by assigning each remaining point to its closest centroid
   - re-compute the centroid of each cluster
3. **until centroids do not change or M iterations reached**

- **the algorithm will always terminate, however it does not always find the optimal solution**
- **this is an example of a greedy algorithm**

Copyright Ellis Horowitz, 2011-2022

K-Means Depends on Centroids

- **Assumes instances are real-valued vectors**
  - **Let $\vec{x}$ represent the vectors in a cluster *c***
- **Then we define the *centroids, (or center of gravity)*, of the cluster to be the mean of the vectors in the cluster; we write this in the following way**

$$\vec{\mu}(c) = \frac{1}{|c|}\sum_{\vec{x}\in c}\vec{x}$$

- **Reassignment of instances to clusters is based on distance to the current cluster centroids.**

Copyright Ellis Horowitz, 2011-2022

There are Several Possible Distance Metrics

- **Euclidean distance (L_2 norm):**

$$L_2(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$$

- **L_1 norm:**

$$L_1(\vec{x},\vec{y}) = \sum_{i=1}^{m}|x_i - y_i|$$

- **Cosine Similarity (transform to a distance by subtracting from 1):**

$$1 - \frac{\vec{x} \bullet \vec{y}}{|\vec{x}| \cdot |\vec{y}|}$$

Copyright Ellis Horowitz, 2011-2022

Some Adjustments to the Algorithm

- **How to pick the initial cluster means points**
  - **Try multiple runs**
    - Choose different random points as the cluster means and see which yields the best result
  - **Select the original set of means by methods other than random**
    - E.g., pick the most distant (from each other) points as cluster centers (this is called the *k-means*++ algorithm)
- **For termination conditions there are several possibilities, e.g.,**
  - After a fixed number of iterations
  - When the document partition is unchanged
  - When the centroid positions don't change

Copyright Ellis Horowitz, 2011-2022

Time Complexity

- **Computing distance between two vectors is O*(m)* where *m* is the dimensionality of the vectors**
- **Re-assigning *n* vectors to *k* clusters: *O(kn)* distance computations, or *O(knm)***
- **Computing centroids: Each vector gets added once to some centroid: *O(nm)***
- **Assume these two steps are each done once for *i* iterations: *O(iknm)***
- ***Note:***
- ***m is the size of the vector***
- ***n is the number of vectors (items)***
- ***k is the number of clusters***
- ***i depends upon convergence***

Copyright Ellis Horowitz, 2011-2022

K-means Clustering – Summary Details

- **Initial centroids are often chosen randomly**
  - Clusters produced vary from one run to another
- **The centroid is (typically) the mean of the points in the cluster**
- **'Closeness' is measured by cosine similarity, a variation of Euclidean distance**
- **Most of the convergence happens in the first few iterations.**
  - Often the stopping condition is changed to 'Until relatively few points change clusters
- **Complexity is $O(i * k * n * m)$**
  - n = number of points, k = number of clusters, i = number of iterations, m = number of attributes

Copyright Ellis Horowitz, 2011-2022

Additional Evaluation Metrics for K-means Clustering

- ***inertia*** evaluates how far the points are within a cluster, or specifically the sum of distances of all the points within a cluster from the centroid of that cluster

Intra cluster distance

- ***Dunn Index*** takes into account the distance between two clusters. This distance between the centroids of two different clusters is known as **inter-cluster distance**. It is computed as the ratio of the minimum inter-cluster distance and the maximum of the intra-cluster distances

$$\text{Dunn Index} = \frac{\text{min(Inter cluster distance)}}{\text{max(Intra cluster distance)}}$$

- *The larger the min(inter-cluster distance) the farther apart are the clusters; the smaller the max(intra-cluster distance) the more compact are the clusters*

Copyright Ellis Horowitz, 2011-2022

Difficulties with K-Means Clustering

When cluster sizes are very different in size, points in the larger section can be mis-clustered

Original Points

K-means (k = 3)

When the densities of the original points are different the more spreadout points can be mis-clustered

Original Points

K-means (k = 3)

How to Choose the Right Number of Clusters

- **Plot a graph, also known as an elbow curve, where the x-axis will represent the number of clusters and the y-axis will be an evaluation metric.**
- **Let's say we use inertia**

Train your model on 2 clusters
Compute and plot the inertia

Train the model on successively higher clusters

Any number of clusters between 6 and 10 will work

the cluster value where this decrease in inertia value becomes constant can be chosen as the right cluster value for your data.

Copyright Ellis Horowitz, 2011-2022

Evaluating K-Means Clusters

- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

  - x is a data point in cluster C_i and m_i is the representative point for cluster C_i
  - can show that m_i corresponds to the center (mean) of the cluster
  - Given two clusters, we can choose the one with the smallest error
  - One easy way to reduce SSE is to increase K, the number of clusters
    - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

Copyright Ellis Horowitz, 2011-2022

Limitations of K-Means

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes
- K-means has problems when the data contains outliers

Copyright Ellis Horowitz, 2011-2022

Hierarchical Clustering Algorithms

- **Two main types of hierarchical clustering**
  - **Agglomerative:**
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left (bottom-up)
  - **Divisive:**
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters), (top-down)
- **Traditional hierarchical algorithms use a similarity or distance matrix**
  - Merge or split one cluster at a time

Copyright Ellis Horowitz, 2011-2022

Agglomerative Clustering Algorithm - a Bottom Up Approach

- **Basic Agglomerative Clustering Algorithm**
  1. Compute the distance matrix between the input data points (i.e. the distance between all pairs of points)
  2. Let each data point be a cluster unto itself
  3. Repeat
  4. Merge the two closest clusters
  5. Update the distance matrix
  6. Until only a single cluster remains
- **Key operation is the computation of the distance between two clusters**
  - Different definitions of the distance between clusters lead to somewhat different algorithms

Copyright Ellis Horowitz, 2011-2022

How Can We Compute the Distance Between Two Clusters

- As before, the **Centroid** of a cluster is the component-wise average of the vectors in a cluster, which is itself a vector
- Example, the Centroid of (1,2,3); (4,5,6); (7,2,6); is (4,3,5)
- **4 possible ways to compute the distance between two clusters**

1. **Center of Gravity**
   - Compute the distance between the two centroids of the cluster
2. **Average Link**
   - Compute the average distance between all pairs of points across the two clusters
3. **Single Link**
   - Compute the distance between the two closest points in the two clusters, i.e. the most cosine similar
4. **Complete Link**
   - Compute the distance between the furthest points in the two clusters, i.e. the least cosine similar

Copyright Ellis Horowitz, 2011-2022

A Dendrogram is Used to Display Clusters

- A **dendrogram** is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering

original input

clusters

corresponding dendrogram

second row clusters are: {a}, {b c}, {d e} {f}
third row clusters are: {a}, {b c} {d e f}

Copyright Ellis Horowitz, 2011-2022

Hierarchical Agglomerative Clustering

- HAC starts with unclustered data and performs successive pairwise joins among items (or previous clusters) to form larger ones
  - this results in a hierarchy of clusters which can be viewed as a dendrogram
  - Dendrograms are usually drawn as shown below
  - The height of an edge can sometimes refer to the degree of similarity
  - useful in pruning search in a clustered item set, or in browsing clustering results

Copyright Ellis Horowitz, 2011-2022

Hierarchical Agglomerative Clustering

- **Basic procedure**
  - **1.** Place each of N documents into a class of its own.
  - 2. Compute all pairwise document-document similarity coefficients
    - ***Total of N(N-1)/2 coefficients***
  - **3. Form a new cluster by combining the most similar pair of current clusters *i* and *j***
    - (use one of the methods described in a previous slide, e.g., complete link, etc.);
    - update similarity matrix by deleting the rows and columns corresponding to *i* and *j*;
    - calculate the entries in the row corresponding to the new cluster *i*+*j*.
  - **4. Repeat step 3 if the number of clusters left is great than 1.**

Copyright Ellis Horowitz, 2011-2022

Example of HAC Input/ Initial setting

- **Start with clusters of individual points and a distance/proximity matrix**

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						

Distance/Proximity Matrix

Copyright Ellis Horowitz, 2011-2022

Intermediate State

- After some merging steps, we have some clusters

	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/Proximity Matrix (above)
Dendrogram (below)

Copyright Ellis Horowitz, 2011-2022

Intermediate State

- **Merge the two closest clusters (C2 and C5) and update the distance matrix.**

	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/Proximity Matrix

Copyright Ellis Horowitz, 2011-2023

How to Define Inter-Cluster Similarity

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

Look back at slide 38

Copyright Ellis Horowitz, 2011-2022

How to Define Inter-Cluster Similarity

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

Copyright Ellis Horowitz, 2011-2022

How to Define Inter-Cluster Similarity

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

Copyright Ellis Horowitz, 2011-2022

How to Define Inter-Cluster Similarity

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

Copyright Ellis Horowitz, 2011-2022

How to Define Inter-Cluster Similarity

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids

Copyright Ellis Horowitz, 2011-2022

General Hierarchical Agglomerative Clustering Algorithm and Complexity

1. **Compute similarity between all pairs of documents**

   $O(N^2)$

2. **Do N – 1 times**
   1. **Find closest pair of documents/clusters to merge**

      Naïve: $O(N^2)$ Priority Queue: $O(N)$ Single link: $O(N)$

   1. **Update similarity of all documents/clusters to new cluster**

      Naïve: $O(N)$ Priority Queue: $O(N \log N)$ Single link: $O(N)$

Copyright Ellis Horowitz, 2011-2022

Divisive Clustering Algorithm

1. Start at the top with all documents in one cluster.
2. The cluster is split using a partitioning clustering algorithm.
   - Use the k-means clustering algorithm, which is linear in computing time whereas HAC (hierarchical agglomerative clustering) algorithms are quadratic
3. Apply the procedure recursively until each document is in its own singleton cluster

- Studies show that the divisive algorithms produce more accurate hierarchies than bottom up
  - Bottom-up methods make clustering decisions based on local patterns without initially taking into account the global distribution. These early decisions cannot be undone.
  - Top-down clustering benefits from complete information about the global distribution when making top-level partitioning decisions.

Copyright Ellis Horowitz, 2011-2022

Divisive Clustering Example

1. Initially, all points in the dataset belong to one single cluster.
2. Partition the cluster into two least similar cluster
3. Proceed recursively to form new clusters until the desired number of clusters is obtained.

All points in one cluster

Two clusters (blue/red)

Three clusters (blue/red/green)

- **At this point the sum of inertia within each of the three clusters is smaller than the previous two examples of two clusters and one cluster**
- **subsequent splitting will only divide points within the existing three clusters**

Copyright Ellis Horowitz, 2011-2022

How to Label Clusters Two Approaches

1. **Show titles of typical documents**
   - Titles are easy to scan
   - Authors create them for quick scanning!
   - But you can only show a few titles which may not fully represent cluster
2. **Show words/phrases prominent in cluster**
   - Use distinguishing words/phrases
   - But harder to scan

- **Common heuristics - list 5-10 most frequent terms in the centroid vector**
  - Drop stop-words;

Copyright Ellis Horowitz, 2011-2022

Search Engine
Question Answering

Copyright Ellis Horowitz 2011-2022

Information Retrieval v. Question Answering

- The name **"information retrieval"** is standard, but as traditionally practiced, it's not really right
- In the past all we got was ***document retrieval***, and beyond that the job is up to us
  - Modern search engines now do a lot better
- They combine information retrieval, knowledge graphs and inferencing, past query history, location, natural language processing and many other criteria
- **Question Answering** (**QA**) is concerned with building systems that automatically answer questions posed by humans in a natural language

Copyright Ellis Horowitz, 2011-2022

People *want* to ask questions…

Examples from Ask.com query log

how much should I weigh
what does my name mean
how to get pregnant
where can I find pictures of hairstyles
who is the richest man in the world
what is the meaning of life
why is the sky blue
what is the difference between white eggs and brown eggs
can you drink milk after the expiration date
what is true love
what is the jonas brothers address

Around 10-20% of query logs are questions such as these

Copyright Ellis Horowitz 2011-2022

Question: *Who was the prime minister of Australia during the Great Depression?*

Answer: *James Scullin (Labor) 1929–31*

How Google used to respond to questions

Page about Curtin (WW II Labor Prime Minister) (Can deduce answer)

Page about Curtin (WW II Labor Prime Minister) (Lacks answer)

Page about Chifley (Labor Prime Minister) (Can deduce answer)

Question: *Who was the prime minister of Australia during the Great Depression?*

Answer: *James Scullin (Labor) 1929–31*

Google's result today

Google has Improved Its Ability to Answer Many Questions

Some Questions are Easily Answered

Copyright Ellis Horowitz 2011-2022

The Original Google Approach

- **Take the question and try to find it as a string on the web**
- **Return the next sentence on that web page as the answer**
- **Works brilliantly if this exact question appears as a FAQ question, etc.**
- **Works poorly most of the time**
- **But a more sophisticated version of this approach has been introduced in recent years combining**
  - Knowledge graph
  - N-grams
  - WordNet
  - NLP techniques

Copyright Ellis Horowitz, 2011-2022

Many Questions Pose Semantic Difficulties

- Who is Michael Jordan?
  - Michael Jordan the basketball player or the Machine Learning guy?
- Key requirement is that entities get identified and disambiguated

Copyright Ellis Horowitz, 2011-2022

Why Natural Language Processing is Required

- **Question: "When was Wendy's founded?"**

- **Passage candidate:**
  - **"The renowned Murano glassmaking industry, on an island in the Venetian lagoon, has gone through several reincarnations since it was founded in 1291. Three exhibitions of 20th-century Murano glass are coming up in New York. By Wendy Moonan."**
- **Answer: 20th Century**
- ***the candidate passage below includes the relevant keywords (Wendy's, founded), but provides an incorrect answer***

Copyright Ellis Horowitz, 2011-2022

More NLP Challenges
Predicate-Argument Structure

- **Q336:** ***When was Microsoft established?***
- **Difficult because Microsoft tends to establish lots of things…**

  <u>Microsoft</u> plans to <u>establish</u> manufacturing partnerships in Brazil and Mexico in <u>May</u>.

- **Need to be able to detect sentences in which `Microsoft' is object of `establish' or close synonym.**
- **A correct result might *not* even include the query term:**

  Microsoft Corp was founded in the US in 1975, incorporated in 1981, and established in the UK in 1982.

NLP: Natural language processing

Copyright Ellis Horowitz, 2011-2022

Some Questions Require Inferences

What is the distance between the largest city in California and the largest city in Nevada?

Google does poorly on this query, misinterpreting Nevada as Nevada County, California

ps: Try the query in Google today to see if they have improved their answer

Copyright Ellis Horowitz 2011-2022

In Some Cases the Data May Not Exist

how many Ph.D. degrees in mathematics were granted by European universities in 1986?

All results are irrelevant

a more recent result; still no relevant links

ght Ellis Horowitz 2011-2022

Some Popular Products Designed for Question/Answering

- **Approach 1: used by Siri**
  - map to known entities and use existing databases over the internet
- **Approach 2: used by Ask.com**
  - detect question type and use a search engine's snippet results
- **Approach 3: used by IBM's Watson**
  - combine approaches 1 and 2
- **Approach 4: used by Google's Knowledge Graph**
  - use an entity - relationship graph that is built in-house and infer the answer

Copyright Ellis Horowitz, 2011-2022

Approach used by Siri: Knowledge-Based Approach

- **Siri was begun as a DARPA project called CALO/PAL (Personalized Assistant that Learns)**

1. **First your voice query is put through a recognizer and a language model and Siri comes up with an interpretation of what was said**
2. **Second Siri builds a semantic representation of the query**
   - Extract times, dates, locations, entities, numeric quantities, and especially user actions (e.g. schedule a meeting, set my alarm, etc)
3. **Siri maps from this semantics to query structured data or resources including:**
   - Geospatial databases
   - Ontologies (Wikipedia infoboxes, Freebase, WordNet, Yago)
   - Restaurant review sources and reservation services (Yelp)
   - Scientific databases (Wolfram Alpha)
   - Conventional search engines (Google, Bing)
4. **Siri then transforms the output above into natural language and then transforms the text back to speech**

Copyright Ellis Horowitz, 2011-2022

Context and Conversation in Virtual Assistants like Siri

- Coreference helps resolve ambiguities
- U: “book a table at Il Fornaio at 7:00 with my mom”
- U: “also send her an email reminder”
- “her” refers to “my mom”; 7:00 refers to pm, not am
- Clarification questions:
- U: “chicago pizza”
- S: “Did you mean pizza restaurants in Chicago or Chicago-style pizza?”

Copyright Ellis Horowitz, 2011-2022

IBM's WATSON System

- Watson was created as a question answering (QA) computing system that IBM built to apply advanced natural language processing, information retrieval, knowledge representation, automated reasoning, and machine learning technologies to the field of open domain question answering
- Watson won a contest on the program Jeopardy against the best human winners
  - https://www.nytimes.com/2011/02/17/science/17jeopardy-watson.html
- However, it has failed to meet earlier expectations, see
  - https://www.nytimes.com/2021/07/16/technology/what-happened-ibm-watson.html
- The Allen Institute for Artificial Intelligence compared Watson's performance on standard natural language tasks like identifying persons, places and the sentiment of a sentence with the A.I. services offered by the big tech cloud providers — Amazon, Microsoft and Google. Watson did as well as, and sometimes better than, the big three
- IBM's Watson in 3 minutes,
  - https://www.techrepublic.com/article/ibm-watson-the-smart-persons-guide/

Copyright Ellis Horowitz, 2011-2022

AskJeeves (now Ask.com)

- **Earlier AskJeeves.com** was well-known as a search engine specializing in Questions/Answers
- Though it still exists, it performs far weaker than sites such as Google

How old is Mariah Carey
Snapshot taken 04/2022

How tall is mount Everest
Snapshot taken 04/2022

Copyright Ellis Horowitz 2011-2022

Question Types: Many Questions Fall into Distinct Categories

Who	Person, Organization
When	Date, Year
Where	Location
In What	Location
How many	Number

Copyright Ellis Horowitz, 2011-2022

3 Main Phases for Question/Answering

1. **QUESTION PROCESSING**
   - Detect question type (who, what, when, where, etc)
   - Identify important entities and formulate queries to send to a search engine
2. **PASSAGE RETRIEVAL**
   - Retrieve ranked documents (snippets only)
   - Break into suitable passages and match against entities
3. **ANSWER PROCESSING**
   - Extract candidate answers (as named entities)
   - Rank candidates
     - **using evidence from relations in the text and external sources**

Copyright Ellis Horowitz, 2011-2022

Question Answering
3 Phase Block Architecture

Definitions:
NL: Natural language
NER: Named Entity Recognition

Copyright Ellis Horowitz 2011-2022

Question Taxonomy

Researchers have tried to organize all question types, e.g.

Copyright Ellis Horowitz, 2011-2022

However Question Taxonomies Can Get Very Large

- **Factoid Questions**
  - Who, where, when, how many
  - The answers fall into a limited and somewhat predictable set of categories, see e.g. to the right
- This set is from a paper by Li and Roth, 2005, *Learning Question Classifiers*
- Their major categories include
  - Abbreviation
  - Description
  - Entity
  - Human
  - Location
  - Numeric
- Questions can be labeled at one or two levels, e.g. either as Numeric or as Numeric:date or as Location or Location:mountain
- This approach argues for a template applied to the query

Tag	Example
ABBREVIATION	
abb	What's the abbreviation for limited partnership?
exp	What does the "c" stand for in the equation E=mc2?
DESCRIPTION	
definition	What are tannins?
description	What are the words to the Canadian National anthem?
manner	How can you get rust stains out of clothing?
reason	What caused the Titanic to sink ?
ENTITY	
animal	What are the names of Odin's ravens?
body	What part of your body contains the corpus callosum?
color	What colors make up a rainbow ?
creative	In what book can I find the story of Aladdin?
currency	What currency is used in China?
disease/medicine	What does Salk vaccine prevent?
event	What war involved the battle of Chapultepec?
food	What kind of nuts are used in marzipan?
instrument	What instrument does Max Roach play?
lang	What's the official language of Algeria?
letter	What letter appears on the cold-water tap in Spain?
other	What is the name of King Arthur's sword?
plant	What are some fragrant white climbing roses?
product	What is the fastest computer?
religion	What religion has the most members?
sport	What was the name of the ball game played by the Mayans?
substance	What fuel do airplanes use?
symbol	What is the chemical symbol for nitrogen?
technique	What is the best way to remove wallpaper?
term	How do you say " Grandma " in Irish?
vehicle	What was the name of Captain Bligh's ship?
word	What's the singular of dice?

Copyright Ellis Horowitz, 2011-2022

More Question Types and Examples

HUMAN	
description	Who was Confucius?
group	What are the major companies that are part of Dow Jones?
ind	Who was the first Russian astronaut to do a spacewalk?
title	What was Queen Victoria's title regarding India?
LOCATION	
city	What's the oldest capital city in the Americas?
country	What country borders the most others?
mountain	What is the highest peak in Africa?
other	What river runs through Liverpool?
state	What states do not have state income tax?
NUMERIC	
code	What is the telephone number for the University of Colorado?
count	About how many soldiers died in World War II?
date	What is the date of Boxing Day?
distance	How long was Mao's 1930s Long March?
money	How much did a McDonald's hamburger cost in 1963?
order	Where does Shanghai rank among world cities in population?
other	What is the population of Mexico?
period	What was the average life expectancy during the Stone Age?
percent	What fraction of a beaver's life is spent swimming?
speed	What is the speed of the Mississippi River?
temp	How fast must a spacecraft travel to escape Earth's gravity?
size	What is the size of Argentina?
weight	How many pounds are there in a stone?

Copyright Ellis Horowitz, 2011-2022

Some General Capabilities for Question-Answering Systems

- **Part-of-Speech Tagging**
  - a piece of software that reads text in some language and assigns **parts** of **speech** to each word, such as noun, verb, adjective, etc.
  - Markov Models are now the standard method for part-of-speech assignment
  - Some current major algorithms for part-of-speech tagging include the Viterbi algorithm, Brill tagger, and Baum-Welch algorithm, see https://en.wikipedia.org/wiki/Part-of-speech_tagging
- **Named Entity Extraction**
  - Software that seeks to locate and classify **named entities** in text into pre-defined categories such as the **names** of persons, organizations, locations, expressions of times, quantities ...
  - See https://en.wikipedia.org/wiki/Named-entity_recognition
- **Determining Semantic Relations**
  - **semantic relations are** concepts or meanings between entities. E.g. [School] is a kind of [educational institution]
    - Opportunity to use WordNet, https://wordnet.princeton.edu/
- **Dictionaries/Thesauri**

Copyright Ellis Horowitz, 2011-2022

Question Processing Tool
Part-of-Speech Recognizer

WP:Wh-pronoun (who/what/when/where)
VBD:Verb, past tense
DT:Determiner
JJ:Adjective
NNP:proper noun, singular
VB:verb
IN:preposition
NN:noun, singular
NNP:proper noun singular
VBZ:verb, third person singular

The two-letter abbreviations are "tags" that are used in POS programs; e.g. see https://pythonprogramming.net/part-of-speech-tagging-nltk-tutorial/

Copyright Ellis Horowitz 2011-2022

Question Processing Tool
Named Entity Recognizer Example

contentSkip to site indexPoliticsSubscribeLog InSubscribeLog InToday's PaperAdvertisementSupported ORG byF.B.I. Agent Peter Strzok PERSON , Who Criticized Trump PERSON in Texts, Is FiredImagePeter Strzok, a top F.B.I. GPE counterintelligence agent who was taken off the special counsel investigation after his disparaging texts about President Trump PERSON were uncovered, was fired. CreditT.J. Kirkpatrick PERSON for The New York TimesBy Adam Goldman ORG and Michael S. SchmidtAug PERSON . 13 CARDINAL , 2018WASHINGTON CARDINAL — Peter Strzok PERSON , the F.B.I. GPE senior counterintelligence agent who disparaged President Trump PERSON in inflammatory text messages and helped oversee the Hillary Clinton PERSON email and Russia GPE investigations, has been fired for violating bureau policies, Mr. Strzok PERSON 's lawyer said Monday DATE .Mr. Trump and his allies seized on the texts — exchanged during the 2016 DATE campaign with a former F.B.I. GPE lawyer, Lisa Page — in PERSON assailing the Russia GPE investigation as an illegitimate "witch hunt." Mr. Strzok PERSON , who rose over 20 years DATE at the F.B.I. GPE to become one of its most experienced counterintelligence agents, was a key figure in the early months DATE of the inquiry.Along with writing the texts, Mr. Strzok PERSON was accused of sending a highly sensitive search warrant to his personal email account.The F.B.I. GPE had been under immense political pressure by Mr. Trump PERSON to dismiss Mr. Strzok PERSON , who was removed last summer DATE from the staff of the special counsel, Robert S. Mueller III PERSON . The president has repeatedly denounced Mr. Strzok PERSON in posts on

- The process of recognizing information units like names, including persons, organizations, location names, and numeric expressions including time, date, money and percent expressions from unstructured text.
- This is an example of supervised learning as training sets are first created
- See https://nlp.stanford.edu/software/ for a Java program and explanation

Copyright Ellis Horowitz 2011-2022

Example of NLP Extraction Used to Build a Knowledge Graph:

Nouns are (usually) entities and verbs are (usually) relationships among the entities and adjectives (usually) describe the relationships

https://kgtutorial.github.io/wsdm-slides/Part2_knowledge-extraction.pdf

Copyright Ellis Horowitz 2011-2022

NER Example and Its Translation Jeopardy Example

The Jeopardy query:

Category: Poets and Poetry: ***GEO***

He was a bank clerk in the Yukon before he published "Songs of a Sourdough" in 1907.

YEAR

Produces the logic formula:

authorof(focus,"Songs of a Sourdough") ^
publish (e1, he, "Songs of a Sourdough") ^
in (e2, e1, 1907) ^ temporallink(publish(...), 1907)

Extracting Candidate Answers from Triple Stores

- **Once we extract a relation from the question, e.g.**

  … he published "Songs of a sourdough"

  `(author-of ?x "Songs of a sourdough")`

- **Many information sources support querying via a triple store**
  - **Wikipedia infoboxes, DBpedia, FreeBase, etc.**
  - **author-of("Songs of a Sourdough", "Robert Service")**

Copyright Ellis Horowitz, 2011-2022

General Keyword Selection Algorithm

1. Use the part-of-speech recognizer to identify all
   - nouns
   - verbs
   - non-stopwords in quotations
   - NNP (proper noun) words in recognized named entities
   - complex nominals with their adjectival modifiers
2. Select the answer type word

Copyright Ellis Horowitz, 2011-2022

Expanding the Keyword Set Using Variants

- There are 3 distinct ways to expand the keyword set determined by the keyword selection algorithm
- Morphological variants (having to do with inflections)
  - invented → inventor → inventions
- Lexical variants (similar to synonyms)
  - killer → assassin
  - far → distance
- Semantic variants
  - like → prefer (a more specific or less specific term)

Copyright Ellis Horowitz, 2011-2022

How to Incorporate Lexical Variants Using Hypernims and Hyponims

***Question*: When was the internal combustion engine invented?**
***Answer*: The first internal combustion engine was built in 1867.**
***Lexical chains*:**

(1) invent:v#1 → HYPERNIM → create_by_mental_act:v#1 → HYPERNIM → create:v#1 → HYPONIM → build:v#1

***Question*: How many chromosomes does a human zygote have?**
***Answer*: 46 chromosomes lie in the nucleus of every normal human cell.**
***Lexical chains*:**

(1) zygote:n#1 → HYPERNIM → cell:n#1 → HAS.PART → nucleus:n#1

WordNet provides hypernims and hyponims

Hypernim is a superordinate grouping which includes subordinate groups
e.g. a musical instrument is a hypernim of guitar;
Hyponim is a word or phrase whose semantics is more specific than its hypernim
e.g. purple is a hyponim of color

Copyright Ellis Horowitz, 2011-2022

Use WordNet for Type Identification

We have already seen the use of WordNet, a lexical database of English nouns, verbs, adjectives, adverbs

"What 20th century poet wrote Howl?"

WordNet permits refinement of poet to specific instances

WordNet also helps determine the TYPE of answer we are looking for

Copyright Ellis Horowitz, 2011-2022

Answer Type Taxonomy

- **Use WordNet to merge named entities with the WordNet hierarchy**

Recall:WordNet is a database of English nouns, verbs, adjectives and adverbs grouped into sets of synonyms (synsets) and relations showing super/subordinate relations (hyperonymy/hyponymy)

If you know the answer should be a person WordNet helps determine what sort of person

Copyright Ellis Horowitz, 2011-2022

Another WordNet Example (1 of 2)

This uses the hypernym (… is a kind of …) relationships in WordNet to construct a path between two entities. For example these hypernym relationships for *fish* and *food* are present in WordNet

Copyright Ellis Horowitz, 2011-2022

Another WordNet Example (2 of 2)

- work out all the paths between fish and food using the generated hypernym trees. It turns out there are three distinct paths.
- The shortest path is between the first definition of fish and the definition of food, as shown.
- One measure is Leacock-Chodorow similarity which is defined as:

$$\text{Semantic Similarity} = -\ln\left(\frac{d}{32}\right)$$

- Which would give a score of 2.37.
- Alternatively one can use the reciprocal of the distance, i.e. 1/3.

Part 2: Passage Retrieval

- **Once we have formulated queries using tools like NER, POS, variant expansion and WordNet results**
- **Send queries to a search engine and retrieve snippet results**
- **Filter the results for correct type**
  - use answer type classification
  - Rank passages based on a trained classifier (application of machine learning)
    - **Features:**
      - **Question keywords, Named Entities**
      - **Longest overlapping sequence,**
      - **Shortest keyword-covering span**
      - **N-gram overlap between question and passage**

Copyright Ellis Horowitz, 2011-2022

Passage Scoring Method

- **Focus on the snippets that are returned, the answers must be extracted from them**
- **Passage ordering is performed using a sort that involves three scores:**
  1. The number of words from the question that are recognized and in the same sequence in the snippet window
  2. The number of words that separate the most distant keywords in the snippet window
  3. The number of unmatched keywords in the snippet window

Copyright Ellis Horowitz, 2011-2022

Ranking Candidate Answers

Q066: Name the first private citizen to fly in space.

- Answer type: Person
- Text passage:

"Among them was Christa McAuliffe, the first private citizen to fly in space. Karen Allen, best known for her starring role in "Raiders of the Lost Ark", plays McAuliffe. Brian Kerwin is featured as shuttle pilot Mike Smith..."

Scoring

There are five words from the question "the first private citizen space"
The answer is adjacent to "the first private citizen. . . "
There are no unmatched keywords in "the first private citizen. . . "

Copyright Ellis Horowitz, 2011-2022

Ranking Candidate Answers

Q066: Name the first private citizen to fly in space.

- Answer type: Person
- Text passage:

"Among them was Christa McAuliffe, the first private citizen to fly in space. Karen Allen, best known for her starring role in "Raiders of the Lost Ark", plays McAuliffe. Brian Kerwin is featured as shuttle pilot Mike Smith..."

- Best candidate answer: Christa McAuliffe

Comments on Scoring this Passage

- Karen Allen is rejected as an answer as it does not occur in the sentence "the first private citizen. . . "
- Brian Kerwin is rejected as the name is far away from "the first private citizen. . . "

Copyright Ellis Horowitz, 2011-2022

Local Alignment Example (1 of 7)

Who shot Kennedy?

Jack assassinated **Oswald**, the man who shot Kennedy, and was **Mrs. Ruby's** Husband.

Three Potential Candidates by type

WHO indicates we are looking for a PERSON

Copyright Ellis Horowitz, 2011-2022

Local Alignment Example (2 of 7)

Question Head word → ***Who shot Kennedy?***

Jack assassinated **Oswald**, the man who shot Kennedy, and was **Mrs. Ruby's** Husband.

"shot Kennedy"
gives us a verb and matching context

Copyright Ellis Horowitz, 2011-2022

Local Alignment Example (3 of 7)

Who shot Kennedy?

Anchor word

Jack assassinated **Oswald**, the man who shot Kennedy, and was **Mrs. Ruby's** Husband.

Look for phrases such as "X shot Kennedy"
or "Kennedy was shot by X"

Copyright Ellis Horowitz, 2011-2022

Local Alignment (4 of 7)

In principle it can be anyone of the three people identified

Who shot Kennedy?

Potential alignments

Jack assassinated **Oswald**, the man who shot Kennedy, and was **Mrs. Ruby's** Husband.

Copyright Ellis Horowitz, 2011-2022

Local Alignment Example (5 of 7)

Who shot Kennedy?

One Alignment: does the "who" refer to Mrs. Ruby?

Jack assassinated **Oswald**, the man who shot Kennedy, and was **Mrs. Ruby's** Husband.

Three Alignment Features :

Copyright Ellis Horowitz, 2011-2022

Local Alignment Example (6 of 7)

1

Who shot Kennedy?

The distance between the question head word "who" and the anchor word Kennedy is 1

One Alignment : does the "who" refer to Mrs. Ruby?
The distance from Kennedy to Mrs. Ruby

Jack assassinated **Oswald**, the man who shot Kennedy, and was **Mrs. Ruby's** Husband.

2

Three Alignment Features :

1. Distance between Question Head word ("who") and the Anchor word ("Kennedy") in the sentence

Copyright Ellis Horowitz, 2011-2022

Local Alignment Example (7 of 7)

Who shot Kennedy?

Jack assassinated **Oswald**, the man who shot Kennedy, and was **Mrs. Ruby's** Husband.

Oswald is properly aligned with "who shot Kennedy" :

Copyright Ellis Horowitz, 2011-2022

A Refined Ranking Scheme

- **Refining the Passage Scoring Method, we can use supervised machine learning to rank the candidate passages according to six criteria**

1. The number of named entities of the right type in the passage
2. The number of question keywords in the passage
3. The longest exact sequence of question keywords that occurs in the passage
4. The rank of the document from which the passage was extracted
5. The proximity of the keywords from the original query to each other. For each passage identify the shortest span that covers the keywords contained in that passage. Prefer smaller spans that include more keywords
6. The N-gram overlap between the passage and the question; Count the N-grams in the question and the N-grams in the answer passages. Prefer the passages with higher N-gram overlap with the question

Copyright Ellis Horowitz, 2011-2022

What is BERT

- ***B***idirectional ***E***ncoder ***R***epresentations from ***T***ransformers
- In 2018, Google introduced and open-sourced BERT
  - BERT is designed to help computers understand the meaning of ambiguous language in text by using surrounding text to establish context
  - It achieved strong results on problems such as sentiment analysis, semantic role labeling, sentence classification and the disambiguation of polysemous words, or words with multiple meanings
- In October 2019, Google announced that they would begin applying BERT to their United States based production search algorithms
- BERT was pre-trained using only an unlabeled, plain text corpus (namely the entirety of the English Wikipedia)
- It continues to learn unsupervised from the unlabeled text and improve even as its being used in practical applications
- the BERT Github repository is available on github,
  git clone https://github.com/google-research/bert.git

Copyright Ellis Horowitz, 2011-2022

Why do we need BERT

- Proper language representation is key for general-purpose language understanding by machines.
- ***Context-free models*** such as word2vec or GloVe generate a single word embedding representation for each word in the vocabulary.
  - For example, the word "bank" would have the same representation in "bank deposit" and in "riverbank".
- ***Contextual models*** instead generate a representation of each word that is based on the other words in the sentence.
  - BERT is a contextual model that captures these relationships in a bidirectional way.

Copyright Ellis Horowitz, 2011-2022

How BERT works

- BERT functions by reading ***bidirectionally***, accounting for the effect of all other words in a sentence on the focus word and eliminating the left-to-right momentum that biases words towards a certain meaning as a sentence progresses.

- At the right BERT is determining which prior word in the sentence the word "it" is referring to, and then using its attention mechanism to weigh the options. The word with the highest calculated score is deemed the correct association (i.e., "it" refers to ”street", not ”animal").

- If this phrase was a search query, the results would reflect this subtler, more precise understanding the BERT reached.

https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model
https://csci572.com/papers/BERT.pdf (the original paper)

Copyright Ellis Horowitz 2011-2022

BERT Has Many Pre-Trained Models

- BERT was trained on Wikipedia and other datasets
- To the right you can see a diagram of additional variants of BERT pre-trained on specialized corpora
- Here are 4 examples of BERT on pre-trained data:
- patentBERT - a BERT model fine-tuned to perform patent classification.
- docBERT - a BERT model fine-tuned for document classification.
- bioBERT - a pre-trained biomedical language representation model for biomedical text mining.
- VideoBERT - a joint visual-linguistic model for process unsupervised learning of an abundance of unlabeled data on Youtube.

https://towardsdatascience.com/bert-for-dummies-step-by-step-tutorial-fb90890ffe03

Copyright Ellis Horowitz, 2011-2022

Microsoft's AskMSR Answering System

- AskMSR is a question answering system developed at Microsoft
- Rather than doing sophisticated linguistic analyses it relies upon information scattered around the web
- AskMSR system stressed how much could be achieved by very simple methods

Microsoft Store Products Support Search Microsoft Research

Research Research areas Products & Downloads Programs & Events People Careers About

An Analysis of the AskMSR Question-Answering System

January 1, 2002

Download PDF

BibTex

Authors

Eric Brill
Susan Dumais
Michele Banko

Published In

Proceedings of EMNLP 2002

Abstract Related Info

We describe the architecture of the AskMSR question answering system and systematically evaluate contributions of different system components to accuracy. The system differs from most question answering systems in its dependency on data redundancy rather than sophisticated linguistic analyses of either questions or candidate answers. Because a wrong answer is often worse than no answer, we also explore strategies for predicting when the question answering system is likely to give an incorrect answer.

http://research.microsoft.com/en-us/um/people/sdumais/EMNLP_Final.pdf

Copyright Ellis Horowitz, 2011-2022

AskMSR: Details

Copyright Ellis Horowitz, 2011-2022

AskMSR: Step 1:Query Rewriting

- **Classify question into categories**
  - **Who is/was/are/were…?**
  - **When is/did/will/are/were …?**
  - **Where is/are/were …?**

a. Category-specific transformation rules

eg "For Where questions, move 'is' to all possible locations"

"Where is the Louvre Museum located"

→ "is the Louvre Museum located"
→ "the is Louvre Museum located"
→ "the Louvre is Museum located"
→ "the Louvre Museum is located"
→ "the Louvre Museum located is"

Nonsense, but who cares? It's only a few more queries to Google.

b. Expected answer "Datatype" (eg, Date, Person, Location, …)

Where is the Louvre Museum located → Place

When was the French Revolution? → Date

Copyright Ellis Horowitz, 2011-2022

AskMSR: Step 2: Query Search Engine

- Send all rewrites to a Web search engine
- Retrieve top N answers (100?)
- For speed, rely just on search engine's "snippets", not the full text of the actual document

Copyright Ellis Horowitz, 2011-2022

AskMSR: Step 3:Mining N-Grams

- **Simple: Enumerate all N-grams (N=1,2,3 say) in all retrieved snippets**
  - Use hash table and other data structures to make this efficient
- **Weight of an n-gram: occurrence count, each weighted by "reliability" (weight) of rewrite that fetched the document**
- **Example: "Who created the character of Scrooge?"**
- **Below are the weights produced by matching the example against a set of N-grams in the N-gram database of the search engine**
  - Dickens - 117
  - Christmas Carol - 78
  - Charles Dickens - 75
  - Disney - 72
  - Carl Banks - 54
  - A Christmas - 41
  - Christmas Carol - 45
  - Uncle - 31

Copyright Ellis Horowitz, 2011-2022

AskMSR: Step 4: Filtering N-Grams

- Each question type is associated with one or more "data-type filters" = regular expression
- When… → Date
- Where… → Location
- What … → Location, Date
- Who … → Person

- Boost score of n-grams that do match the regular expression
- Lower score of n-grams that don't match the regular expression

Copyright Ellis Horowitz, 2011-2022

AskMSR: 5: Tiling the Answers

Copyright Ellis Horowitz, 2011-2022

Introduction to

Relevance Feedback
Query Expansion
And
Classification

Some slides are from Helen Yannakoudakis and Ronan Cummins of Univ. of Cambridge

Copyright Ellis Horowitz, 2011-2022

Outline

- Examples of Relevance Feedback
- Query Expansion examples and techniques for producing relevance feedback
- Rocchio Algorithm for Relevance Feedback (under ideal conditions, i.e. relevant and non-relevant documents are known)
- Rocchio Algorithm for Classification including an online version
- K-Nearest Neighbor Algorithm for Classification

Copyright Ellis Horowitz, 2011-2022

Relevance Feedback

- ***Idea*: The idea is to involve the user in the retrieval process by having the user give feedback about which results are most relevant**
- **The idealized process of user feedback of relevance is to improve recall**
  - User issues a (short, simple) query
  - The user marks some results as relevant/non-relevant.
  - The system computes a better representation of the information need based on feedback.
  - New results have (hopefully) better recall.
  - Relevance feedback can go through one or more iterations
- **E.g. Google and Bing proivde relevance feedback without user intervention**
  - However, their RSS feeds permit users to specify relevant and non-relevant results, thus improving the value of the feed

Copyright Ellis Horowitz, 2011-2022

Examples where Relevance Feedback Would be Useful

"politics" unrefined query

"jaguar" ambiguous

"plain fuel"
Two interpretations are possible

"cars" unrefined

- Relevance feedback and query expansion aim to overcome these problems.

Relevance Feedback will Improve Recall

- **Methods for tackling this problem split into two classes**

1. **Local methods:** adjust a query relative to the documents returned (query-time analysis on a portion of documents)
   - **Main local method: either get user input on relevance or use Google's approach of "Did you mean?"**
2. **Global methods:** adjust query based on some global resource, e.g. a thesaurus or knowledge graph (i.e., a resource that is not query dependent)

Copyright Ellis Horowitz, 2011-2022

Google Uses Query Expansion in its Search Results

- **Google uses six techniques for expanding a query to make it more meaningful**

Google's Query Expansion	Explanation and observations
Word stemming	The term searched is reduced to a root or 'stem'- words of the same stem can be ranked for the search containing only one of them.
Acronyms	An abbreviation / acronym / initialism searched is automatically resolved to the full version. If there are several possible variants of one and the same acronym, Google will do its best to **mention all variations on the first page** in the following order: (1) most popular and hence most probable one (2) all the rest.
Misspellings	With misspelled words searched, Google will suggest the correct variant as well as list sites that use that correct variant. Note: based on common webmasters' experience, **Google uses some other (unclear) algorithm for typos**: sites ranked for the misspelled word (using it nowhere on the page or in the text of backlinks) may have low authority and rank nowhere if the correct spelling is searched.
Synonyms	Sometimes Google includes related words in the search results. Very often the query is extended this way when it is **obvious the phrase was used improperly**. The synonym substituting the part of the phrase is **not bolded** as a rule.
Translations	In some instances, Google seems to translate search keywords into other languages and return results from that language.
Ignored words	Occasionally some ("**insignificant**") words appear to have been dropped completely from the query.

Query Expansion Example 1

- Query expansion is used to increase precision and recall.
- The query "airplane" primarily produces a set of movie results

- query expansion which Google calls Feedback will expand the query making use of the knowledge graph

Query Expansion: Example 2

- The query is “palm”
- The carousel of choices includes restaurants, hotels all named “palm”
- The related searches include references to palm trees, palm pdas and palm hotels

Once Again We Look at WordNet to Implement Query Expansion

- **One way to perform query expansion is to use WordNet**
- **A more detailed database of semantic relationships between English words.**
- **Developed by famous cognitive psychologist George Miller and a team at Princeton University.**
- **About 144,000 English words.**
- **Nouns, adjectives, verbs, and adverbs grouped into about 109,000 synonym sets called *synsets*.**

Copyright Ellis Horowitz, 2011-2022

WordNet Synset Relationships

- **Antonym: front → back (opposite)**
- **Attribute: benevolence → good (noun to adjective)**
- **Pertainym: alphabetical → alphabet (relating to; adjective to noun)**
- **Similar: unquestioning → absolute (resembling)**
- **Cause: kill → die**
- **Entailment: breathe → inhale (a necessary part)**
- **Holonym: chapter → text (part to whole)**
- **Meronym: computer → cpu (whole to part)**
- **Hyponym: plant → tree (specialization)**
- **Hypernym: apple → fruit (generalization)**

Copyright Ellis Horowitz, 2011-2022

Query Expansion Using WordNet

- **Form an expanded query by:**

1. **Adding WordNet synonyms in the same synset**
2. **Adding WordNet hyponyms to add specialized terms**
3. **Adding WordNet hypernyms to generalize a query**
4. **Adding other related terms to expand query**

Copyright Ellis Horowitz, 2011-2022

Query Expansion Using Thesaurus

- Another way to do query expansion is to make use of a thesaurus
- For each term, *t*, in a query, expand the query with synonyms and related words of *t* from the thesaurus.
- One can weight added terms less than the original query terms.
- This generally increases recall, but it may significantly decrease precision

Copyright Ellis Horowitz, 2011-2022

Query Expansion Summary of Points

- Query Expansion is transparent in that it allows the user to see (select) expansion terms.
- Use WordNet to identify synonyms and related terms
- Use a thesaurus to develop a co-occurrence of related terms
- Query log mining approaches have also been shown to be useful

- However, query expansion still suffers from problems of polysemy (multiple meanings of a word or phrase)
  - E.g. "bank" is a financial institution, a building housing a financial institution, the process of saving money, a steep slope, something held in reserve, "you can rely (bank) on me"

Copyright Ellis Horowitz, 2011-2022

Rocchio Algorithm: Basics

- The Rocchio algorithm is a method of relevance feedback
- It was initially developed by the SMART Information Retrieval System in 1960-1964.
- It assumes documents are represented using the vector space model
- The algorithm uses the notions of relevant/non-relevant documents and centroids
- Recall: the centroid is the center of mass (or vector average) of a set of points.
- *Definition*: Centroid

$$\vec{\mu}(C) = \frac{1}{|C|}\sum_{d \in C} \vec{d}$$

where C is a set of documents, |C| is the size of the set, and d is the normalized vector representing document d

- **Note**: We have seen centroids before in the k-means algorithm

Copyright Ellis Horowitz 2011-2022

Centroid Example

3 regions each with its own centroid

The centroid may not correspond to a document in your set

The boundary between two classes in Rocchio classification is the set of points with equal distance from the two centroids

Rocchio Algorithm Derivation

Assuming someone has identified the set of relevant (D_r) and non-relevant (D_{nr}) documents, the algorithm aims to find the query q that maximizes similarity with the set of relevant documents D_r while minimizing similarity with the set of non-relevant documents D_{nr} :

$$q_{opt} = \arg\max\, [sim(q, D_r) - sim(q, D_{nr})]$$

Under cosine similarity, the optimal query for separating relevant and non-relevant documents is:

$$q_{opt} = 1/|D_r| \sum_{d_j \text{ in } D_r} d_j - 1/|D_{nr}| \sum_{d_j \text{ in } D_{nr}} d_j$$

which is the vector difference between the centroids of the relevant and non-relevant documents.

Copyright Ellis Horowitz, 2011-2022

Rocchio Algorithm for Relevance Feedback - in Practice

- In practice, however, we usually do not know the full set of relevant and non-relevant sets.
- For example, a user might only label a few documents as relevant / non-relevant.
- Therefore, in practice Rocchio is often parameterised as follows:

$$\vec{q}_m = \alpha \vec{q} + \frac{\beta}{|D_r|} \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \frac{\gamma}{|D_n|} \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

where q is the original query vector; D_r and D_n are the sets of known relevant and non-relevant documents.

α, β, and γ are weight parameters attached to each component.

Reasonable values are $\alpha = 1.0$, $\beta = 0.75$, $\gamma = 0.15$

•Note: if the final vale of q_m has negative term weights, set those to 0.

Copyright Ellis Horowitz, 2011-2022

Rocchio in Practice

- Represent query and documents as weighted vectors (e.g., tf–idf).
- Use Rocchio formula to compute new query vector (given some known relevant / non-relevant documents).
- Calculate cosine similarity between new query vector and the documents.
- Rocchio has been shown useful for increasing both precision and recall because it contains aspects of positive and negative feedback.
- Positive feedback is much more valuable than negative (i.e., indications of what *is* relevant) so typically systems set $\gamma < \beta$ or even $\gamma = 0$.

2D Rocchio Example

- For 2D examples the relevant set is generally much smaller than the non-relevant set;
- As a result we need a slightly modified rule

Let circles represent relevant documents,
Let Xs represent nonrelevant documents

2D Rocchio Illustrated (1 of 9)

$\vec{\mu}_R$: centroid of relevant documents

2D Rocchio Illustrated (2 of 9)

$\vec{\mu}_R$ does not separate relevant / nonrelevant.

2D Rocchio Illustrated (3 of 9)

$\vec{\mu}_{NR}$: centroid of nonrelevant documents.

2D Rocchio Illustrated (4 of 9)

Consider the two centroids

2D Rocchio Illustrated(5 of 9)

$\vec{\mu}_R$ - $\vec{\mu}_{NR}$: centroid difference vector

2D Rocchio Illustrated(6 of 9)

Add difference vector to $\vec{\mu}_R$ …

2D Rocchio Illustrated(7 of 9)

… to get $\vec{q}_{opt}$

2D Rocchio Illustrated(8 of 9)

Note that the boundary computed during the Rocchio algorithm in This case is viewed as a circle;

Tests of new documents are easily determined to either fit within the circle or not

$\vec{q}_{opt}$ now separates relevant / nonrelevant perfectly.

2D Rocchio Illustrated(9 of 9)

$\vec{q}_{opt}$ separates relevant / nonrelevant perfectly.

Rocchio Algorithm Used for Classification

- More typically, the boundary determination in Rocchio is not a circle, but a hyperplane
- Given two centroids of two classes of documents, the boundary between the two classes is the set of points with equal distance from the two centroids
- Once the boundary is determined, the Rocchio rule is to classify a point according to the region it falls into, or equivalently, determining the centroid that the point is closest to

Copyright Ellis Horowitz, 2011-2022

Rocchio Classification Algorithm

TrainRocchio(C, D)

For each c_j *in C*

do $D_j = \{d: <d,c_j> \text{ is in } D\}$

$u_j = (1/|Dj|) * \Sigma d_j$ *for* d_j *in D*

return $\{u_1, \ldots u_j\}$

See Figure 14.4 in our textbook
D is the set of documents
C is the set of classes determined earlier
D_j is computed using Euclidean distance and includes the set of documents in class c_j
u_j is the new centroid for documents in D_j

ApplyRocchio($\{u_1, \ldots u_j\}$, *d*)

return *arg min* $|u_j - d_j|$

For each new document *d* and centroids u_1 to u_j, place *d* in the class whose centroid is the closest to *d*

Copyright Ellis Horowitz, 2011-2022

Computing Time for Rocchio

Mode	Complexity	
Training	$O(\|D\|L_{ave} + \|C\| \, \|V\|)$	← Add all docs to their centroid, And compute the centroid
testing	$O(L_a + \|C\| \, M_a) = O(\|C\| \, M_a)$	← #centroids * #types

- L_{ave} is the average number of tokens per document (*this is small*)
- L_a and M_a are the numbers of tokens and types respectively in the test document
- The time to compute the Euclidean distance between the class centroid and a document is $O(|C|M_a)$ *(the number of centroids times number of tokens)*
- Adding all documents to their respective (unnormalized) centroid is $\Theta(|D|L_{ave})$

(as opposed to $\Theta(|D||V|)$) since we need only consider non-zero entries.

- Dividing each vector sum by the size of its class to compute the centroid is

$\Theta(|V|)$.

See Table 14.2 in our textbook

Conclusion: Overall training time is linear in the size of the collection

Copyright Ellis Horowitz, 2011-2022

Rocchio Example Simple Case

- Given two classes, C1 and C2 and the document vectors are
- C1: (2,7), (3,8), (1,6) (3 documents)
- C2: (6,4), (8,6), (5,2), (9,4) (4 documents)
- Which class, C1 or C2 does the document (74,36) fall into?
- Training phase of Rocchio
  - centroid of C1, ((2+3+1)/3, (7+8+6)/3) = (2,7)
  - centroid of C2 = ((6+8+5+9)/4, (4+6+2+4)/4) = (7,4)
- Actual phase to find into which class a new document fits:
  - compute the distance between the new document and the centroids or alternatively
  - draw the perpendicular divisor between the two centroids and then see where the new vector falls

Copyright Ellis Horowitz, 2011-2022

Rocchio Example Simple Case

- To draw the perpendicular divisor between the two centroids and then see where the new vector falls recall:
  - The decision boundary will be a straight line, the perpendicular bisector of the two centroids;
  - The equation of the line defines the boundary
- Compute the perpendicular bisector between (2,7) and (7,4)
- if m_1 is the slope of the line between the two points, and m_2 is the slope of the perpendicular bisector, then $m_1 m_2 = -1$
- Slope $m_1 = (y_2-y_1)/(x_2-x_1) = (7-4)/(2-7) = -3/5$ and so the slope of m_2 is 5/3
- The line is y = mx + b where m = 5/3 and a point it passes through is the midpoint of the two centroids; the midpoint of the centroids is ((2+7)/2, (7+4)/2) = (9/2, 11/2)
- So the value of b is 11/2 = (5/3)*(9/2) + b or (11/2) – (45/6) = -12/6 = -2
- *y = (5/3)x-2;* returning to the original point *(74,36)* we see it falls into class C2

Copyright Ellis Horowitz, 2011-2022

Rocchio Extension

- **Suppose we have new documents arriving, how can we update the centroids in constant time?**
  - **Suppose we had *N* documents with centroid *(x,y)* and a new document with coordinates *(a, b)* arrives; then the new centroid is:**
  - ***((Nx + a)/(N+1), (Ny+b)/(N+1))***
  - **One multiplication and one addition is needed**
- **Now suppose one document is discarded from the training set; what is the new centroid**
  - ***((Nx-a)/(N-1), (Ny-b)/(N-1))***
- **The online Rocchio algorithm has to constantly update itself as new documents are added and deleted**

Copyright Ellis Horowitz, 2011-2022

kNN - *k* Nearest Neighbor Method

Classification Algorithm

Copyright Ellis Horowitz 2011-2022

Classification is Different from Clustering

- In general, in ***classification*** you have a set of predefined classes and want to know which class a new object (document) belongs to.
- Remember, ***Clustering*** tries to group a set of objects and find whether there is *some* relationship between the objects.
  - we already saw two algorithms for clustering, K-Means Algorithm and Agglomerative Clustering algorithm
- In the context of machine learning, classification is *supervised learning* and clustering is *unsupervised learning*
  - **Clustering** requires a. an algorithm, b. a similarity measure, and c. a number of clusters
  - **classification** has each document labeled in a class and an algorithm that assigns documents to one of the classes

Copyright Ellis Horowitz, 2011-2022

Classification Methods

- **Manual classification**
  - Used by the original Yahoo! Directory
  - **Other search engines did similar things, e.g.**
    - Looksmart, http://www.looksmart.com/
    - Open Directory Project, https://www.dmoz.org/
    - PubMed, http://www.ncbi.nlm.nih.gov/pubmed
  - **Accurate when job is done by experts**
  - **Consistent when the problem size and team is small**
  - **Difficult and expensive to scale**
    - Means we need automatic classification methods for big problems

Copyright Ellis Horowitz, 2011-2022

The Problem Statement for Classification

- **Given two things:**
  1. **A description of an instance, $x \in X$, where X is the *instance language* or *instance space*, and**
  2. **A fixed set of categories:**

     $C = \{c_1, c_2, \ldots, c_n\}$
- **Determine:**
  - **The category of x: $c(x) \in C$, where $c(x)$ is a *categorization function* whose domain is X and whose range is C.**
    - **Functions that categorize are called "classifiers"**

Copyright Ellis Horowitz, 2011-2022

Classification Using Vector Spaces

- **In vector space classification, the training set corresponds to a labeled set of document vectors**
- **Premise 1: Documents in the same class form a contiguous region of space**
  - This is referred to as the *contiguity hypothesis*
- **Premise 2: Documents from different classes don't overlap (much)**
- **Learning a classifier is equivalent to building (defining) surfaces to delineate classes in the space**

Copyright Ellis Horowitz, 2011-2022

Ways to Measure Distance

For normalized vectors Euclidean distance and cosine similarity correspond

Distance functions

Euclidean $$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan $$\sum_{i=1}^{k}|x_i - y_i|$$

Minkowski $$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$$

Copyright Ellis Horowitz, 2011-2022

k Nearest Neighbor Classification Algorithm

- **Initially we assume we have a set of *N* documents that have already been classified**
  - the WDM videos assume the class names are colors (see the Schedule of Lectures)
- **To classify a document *d***
  - locate among the *N* documents the *k* closest ones
  - from these *k* neighbors, pick the class that occurs most often, the majority class, and use that as the label for *d*

Copyright Ellis Horowitz, 2011-2022

Example: *k*=6 (6NN)

5 neighbors are colored red, one is colored green, so the yellow diamond is colored red

When k=1, the document is assigned to its nearest neighbor

Copyright Ellis Horowitz 2011-2022

K-Nearest Neighbor Another Example

- $k = 1$:
  - Belongs to square class
- $k = 3$:
  - Belongs to triangle class
- $k = 7$:
  - Belongs to square class

- **Choosing the value of *k*:**
  - **If *k* is too small, sensitive to noise points**
  - **If *k* is too large, neighborhood may include points from other classes**
  - **Choose an odd value for *k*, to eliminate ties**

Copyright Ellis Horowitz, 2011-2022

Nearest-Neighbor Without Learning

- **Learning: there is no learning step; just store the labeled training examples *D***
- **Testing instance *x (under 1-NN)*:**
  - Compute the distance between *x* and all examples in *D*.
  - Assign *x* the category of the closest example in *D*.
- **Rationale of *k-NN*: contiguity hypothesis**
  - if documents do form contiguous regions in space, then *k-NN* makes sense
- **Does not compute anything beyond storing the examples**
  - we are NOT determining any hierarchy or model
- ***K-NN* has also been called:**
  - Case-based learning
  - Memory-based learning
  - Lazy learning

Copyright Ellis Horowitz, 2011-2022

Choice of *K*

- **The best choice of *k* depends upon the data;**
  - generally, larger values of *k* reduce the effect of noise on the classification, but make boundaries between classes less distinct.
- The accuracy of the *k-NN* algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance
- In binary (two class) classification problems, (e.g. male or female) it is helpful to choose *k* to be an odd number as this avoids tied votes
- Choosing the optimal value for k is best done by first inspecting the data
- Cross-validation is another way to retrospectively determine a good value of K by using an independent dataset to validate the K value
- Historically, the optimal K for most datasets has been between 3-10

Copyright Ellis Horowitz, 2011-2022

Voronoi Diagram

For the k-Nearest Neighbor Algorithm, k = 1 is a special case

When *k* = 1, each training vector defines a region in space, defining a *Voronoi* partition of the space

$$R_i = \{x : d(x, x_i) < d(x, x_j), i \neq j\}$$

R_i* is Region *i

Copyright Ellis Horowitz, 2011-2022

When k=1 – A Special Case

- A **Voronoi diagram** is a partitioning of a plane into regions based on distance to points in a specific subset of the plane
- Decision boundaries in *1-NN* are concatenated segments of a Voronoi tessellation (e.g. polygons)
- The set of points (called class labels) is specified beforehand
- For each class label there is a corresponding region consisting of all points closer to that class label than to any other. These regions are called Voronoi cells

20 points (class labels) and their Voroni regions; Line segments are all points equidistant to three or more regions

Copyright Ellis Horowitz, 2011-2022

K=1 Nearest Neighbor Regions are Polygons

- **For *1-NN* we assign each document to the class of its closest neighbor**
- **For *k-NN* we assign each document to the majority class of its *k* closest neighbors where *k* is a parameter**

The two classes are: X and circle, and the star document is falling into the circle area;
Double lines define the regions in space where documents are similar;
think of each region as defining a cellphone tower

***K-NN* is an example of a non-linear classifier; (Rocchio is a linear classifier)**

Copyright Ellis Horowitz, 2011-2022

Computing Times

- **Training a *kNN* classifier simply consists of determining *k* and pre-processing the documents**
- **If we preselect *k* and do not pre-process, then *kNN* requires no training at all**
- **It makes more sense to pre-process training documents once as part of the training phase rather than repeatedly every time we classify a new test document**

kNN with preprocessing of training set

training $\Theta(|\mathbb{D}|L_{ave})$

testing $\Theta(L_a + |\mathbb{D}|M_{ave}M_a) = \Theta(|\mathbb{D}|M_{ave}M_a)$

kNN without preprocessing of training set

training $\Theta(1)$

testing $\Theta(L_a + |\mathbb{D}|L_{ave}M_a) = \Theta(|\mathbb{D}|L_{ave}M_a)$

Recall from previous slide: L_{ave} is the average number of tokens per document
L_a and M_a are the numbers of tokens and types respectively in the test document

Copyright Ellis Horowitz, 2011-2022

Observations on *k-NN*

- **The K-Nearest-Neighbor model has two major drawbacks**.

1. **performance**.
   - you have to load all of your training data and calculate distances to all training samples; Over a big training set, this is a major drawback.
2. **distance metric**. Using Euclidean distance can be a problem in high dimensions as well as with noisy features

- **To summarize**, if you have a system that has to learn a sophisticated (i.e. nonlinear) pattern with a small number of samples (and usually dimensions), *k-NN* models are usually one of the best options out there.

Copyright Ellis Horowitz, 2011-2022

K-NN: Final Points

- **No feature selection necessary**
- **No training necessary**
- **Scales well with large number of classes**
  - Don't need to train *n* classifiers for *n* classes
- **Classes can influence each other**
  - Small changes to one class can have ripple effect
- **In most cases it's more accurate than Rocchio**

Copyright Ellis Horowitz, 2011-2022

Algorithm Comparison

K-Means	K-Nearest Neighbors
Clustering algorithm	Classification Algorithm
Uses distance from data points to k-centroids to cluster data into k-groups.	Calculates k nearest data points from data point X. Uses these points to determine which class X belongs to
Centroids are not necessarily data points.	"Centroid" is the point X to be classified.
Updates centroid on each pass by calculations over all data in a class.	Data point to be classified remains the same.
Must iterate over data until center point doesn't move.	Only requires k distance calculations.

Copyright Ellis Horowitz, 2011-2022

Recommendation Systems

Collaborative Filtering & Content-Based Recommendations

Copyright Ellis Horowitz 2011-2022

Recommendations are Found Everywhere

Amazon uses recommendations to great effect; here a user looks at a book, *Age of Spiritual Machines* and Amazon's recommendations

Copyright Ellis Horowitz 2011-2022

Value of Recommendation Systems

- **Recommendation Systems have been shown to substantially increase sales at on-line stores**
  - **Recommendation engines can significantly increase revenue, improve CTRs and conversions. It also contributes to the improvement of factors more difficult to measure, such as customer satisfaction, and it increases customer retention.**
  - **https://neoteric.eu/blog/how-to-boost-sales-with-a-recommender-system/**
  - **"YouTube's video recommending algorithm is, literally, worth billions of dollars:" the company's CPO, Neal Mohan, reports that over 70 percent of the time spent on YouTube is spent watching recommended videos**
  - **https://faun.pub/the-algorithm-worth-billions-how-youtubes-addictive-video-recommender-works-d75646dac6a3**

Copyright Ellis Horowitz, 2011-2022

The YouTube Recommendation Algorithm Why All the Fuss

- **"YouTube's recommender AI still a horror show, finds major crowdsourced study"**
- **https://techcrunch.com/2021/07/07/youtubes-recommender-ai-still-a-horrorshow-finds-major-crowdsourced-study/**
- New research published July 2021 by Mozilla backs that notion up, suggesting YouTube's AI continues to puff up piles of "bottom-feeding"/low-grade/divisive/disinforming content — stuff that tries to grab eyeballs by triggering people's sense of outrage, sewing division/polarization or spreading baseless/harmful disinformation — which in turn implies that YouTube's problem with recommending terrible stuff is indeed systemic; a side effect of the platform's rapacious appetite to harvest views to serve ads.

Copyright Ellis Horowitz, 2011-2022

Why Recommendation Systems Are Needed

http://www.wired.com/wired/archive/12.10/tail.html?pg=3

Copyright Ellis Horowitz, 2001-2022

Scarcity versus Abundance

- **Since online systems maintain large quantities of goods, systems that provide recommendations serve an important purpose**
  - In some cases items sold from the long tail, (i.e. those not particularly popular) can cumulatively outweigh the initial portion of the graph, an in effect produce the majority of sales
- **Recommendation systems are expected to increase diversity because they help us discover new products.**
  - However, some algorithms may unintentionally do the opposite.
  - Because they recommend products based on past sales or ratings, they cannot usually recommend products with limited historical data. This can create a rich-get-richer effect for popular products
  - This bias toward popularity can prevent what are otherwise better consumer-product matches.
    - Several collaborative filtering algorithms have been developed to promote diversity and the long tail by recommending novel, unexpected, and serendipitous items
  - See https://en.wikipedia.org/wiki/Collaborative_filtering

Copyright Ellis Horowitz, 2011-2022

Two Types of Recommendation Systems

- A *recommendation system* is any system which provides a recommendation/prediction/opinion to a user on items

1. A classic *content-based filtering system* uses item similarity/clustering to recommend items like ones you like
2. A classic *collaborative filtering system* uses user similarity, i.e. the links between users and the item they chose as the basis of recommendations

- Commonly many companies use *hybrid systems* in which both kinds of techniques are employed

Copyright Ellis Horowitz, 2011-2022

Difference between Collaborative and Content-based Filtering-An Example

- *Here are two early systems that recommended music*
- *Last.fm* creates a "station" of recommended songs by observing what bands and individual tracks ***the user*** has listened to on a regular basis (*user similarity*) and comparing those against the listening behavior of other users.
  - Last.fm will play tracks that do not appear in the user's library, but are often played by other users with similar interests.
  - As this approach leverages the behavior of users, it is an example of a ***collaborative filtering technique***.
- *Pandora* uses the ***properties of a song*** or artist (a subset of the 400 attributes provided by the Music Genome Project) to seed a "station" that plays music with similar properties (*item similarity*).
  - User feedback is used to refine the station's results, deemphasizing certain attributes when a user "dislikes" a particular song and emphasizing other attributes when a user "likes" a song.
  - This is an example of a ***content-based technique***.

Copyright Ellis Horowitz, 2011-2022

Basic Flowchart of a Recommender System

An Example Restaurant Recommendations

- **Suppose we have a list of all Los Angeles restaurants**
  - **with ↑ and ↓ ratings for *some***
  - **as provided by USC students**
- **Which restaurant(s) should I recommend to you?**

Copyright Ellis Horowitz, 2011-2022

Input

Alice	Il Fornaio	Yes
Bob	Ming's	No
Cindy	Straits Café	No
Dave	Ming's	Yes
Alice	Straits Café	No
Estie	Zao	Yes
Cindy	Zao	No
Dave	Brahma Bull	No
Dave	Zao	Yes
Estie	Ming's	Yes
Fred	Brahma Bull	No
Alice	Mango Café	No
Fred	Ramona's	No
Dave	Homma's	Yes
Bob	Higashi West	Yes
Estie	Straits Café	Yes

Copyright Ellis Horowitz 2011-2022

Algorithm 0

- Strategy: Recommend to you the most popular restaurants
  - say # positive votes minus # negative votes
- But this ignores
  - your culinary preferences
  - judgments of those with similar preferences
- How can we exploit the wisdom of "like-minded" people?
- Basic assumption
  - Preferences are not random
  - Assumption: if I like Il Fornaio, it's more likely I will also like Cenzo (another Italian restaurant)

Copyright Ellis Horowitz, 2011-2022

Cast the Input as a Matrix

	Brahma Bull	Higashi West	Mango	Il Fornaio	Zao	Ming's	Ramona's	Straits	Homma's
Alice		Yes	No	Yes				No	
Bob		Yes				No		No	
Cindy				Yes	No			No	
Dave	No			No	Yes	Yes			Yes
Estie				No	Yes	Yes		Yes	
Fred	No						No		

Called a *utility matrix*

Each row represents an individual and each column a restaurant
In this example, entries are either yes/no;
In the more general case they can be any value

Copyright Ellis Horowitz 2011-2022

The Utility Matrix

- In a recommendation-system application there are two classes of entities, which we shall refer to as *users* and *items*
  - Users have preferences for certain items, and these preferences must be teased out of the data.
- The data itself is represented as a ***utility matrix***, giving for each user-item pair, a value that represents what is known about the degree of preference of that user for that item.
- *Values might come from an ordered set*, e.g., integers 1–5 representing the number of stars that the user gave as a rating for that item.
- We assume that the matrix is *sparse*, meaning that most entries are "unknown."
- An unknown rating implies that we have no explicit information about the user's preference for the item.

Copyright Ellis Horowitz, 2011-2022

A New Utility Matrix with Rankings

	Brahma Bull	Higashi West	Mango	Il Fornaio	Zao	Ming's	Ramona's	Straits	Homma's
Alice		1	-1	1				-1	
Bob		1				-1		-1	
Cindy				1	-1			-1	
Dave	-1			-1	1	1			1
Estie				-1	1	1		1	
Fred	-1						-1		

View all other entries as zeros for now.

- To compute the similarity between individual's preference vectors we can use inner products as a good place to start, e.g.
  - Dave has similarity 3 with Estie,
    - e.g. (-1,0,0,-1,1,1,0,0,1) and (0,0,0,-1,1,1,0,1,0)
    - (i.e. there are three matching values of either 1 or -1)
  - but -2 with Cindy (0,0,0,1,-1,0,0,-1,0) (a zero value doesn't count).
- Perhaps recommend Straits Cafe to Dave and Il Fornaio to Bob, etc.

Copyright Ellis Horowitz 2011-2022

Another Utility Matrix Example - Movies

	Avatar	LOTR	MATRIX	PIRATES
ALICE	1		0.2	
BOB		0.5		0.3
CAROL	0.2		1	
DAVID				0.4

- The blank spaces indicate either the user has not seen the movie or has decided not to rate it
- Main issue: how to fill in the values in the missing fields
- In general there are two basic techniques for populating the utility matrix
  1. Ask users to rate items
     E.g. movies, online stores from purchasers
  2. Make inferences from user behaviors
     Assumption: Users who watch a movie must "like" it, so above we can assign it a 1;
     We are mostly interested in fields for which the ratings are high

Copyright Ellis Horowitz, 2011-2022

Recommending Documents Also can be Viewed as a Form of Recommendation System

- **If the items we are considering are documents, then the profile will be the set of "important" words in the document**
- **How do we pick important words**
  - **We use the TF-IDF formulation seen earlier**

Profile of a document d_j is the vector of weights $w_{i,j}$ $Content(d_j) = (w_{1j}, \ldots w_{kj})$.

$$w_{i,j} = TF_{i,j} \times IDF_i \qquad TF_{i,j} = \frac{f_{i,j}}{\max_z f_{z,j}}, \qquad IDF_i = \log \frac{N}{n_i}.$$

- ***TF* : Term Frequency, IDF : Inverse Document Frequency**
- ***N* : Number of the documents**
- **n_i : How many times an element is seen in all of the documents**
- **$f_{i,j}$: Number of times an element is seen in the document d_j**

Copyright Ellis Horowitz, 2011-2022

Example 1
Boolean Utility Matrix

- **Items are movies, only feature is Actor**
  - **Item profile: vector with *0* or *1* for each actor**
- **Suppose user *X* has watched *5* movies**
  - ***2* movies featuring actor A (movies 1 and 3)**
  - ***3* movies featuring actor B (movies 2, 4, and 5)**
- **User profile = mean of item profiles**
  - **Feature A's weight = *2/5 = 0.4***
  - **Feature B's weight = *3/5 = 0.6***

	(ActA,	ActB,	ActC,	ActD,	ActE)
Movie1(	1,	0,	0,	0,	0)
Movie2(	0,	1,	0,	0,	0,)
Movie3(	1,	0,	0,	0,	0)
Movie4(	0,	1,	0,	0,	0)
Movie5(	0,	1,	0,	0,	0)

ActA's weight = Sum(ActA)/5

Copyright Ellis Horowitz, 2011-2022

Example 2
Star Ratings Normalized

- **Same example, 1-5 star ratings**
  - **Actor A's movies rated *3* and *5***
  - **Actor B's movies rated *1, 2,* and *4* (note: *1* and *2* are negative ratings)**
- **Useful step: normalize ratings by subtracting user's mean rating (3)**
  - **Given ratings 1,2,3,4,5 – the mean is 3, so:**
  - **Actor A's normalized ratings = 0, +2**
    - **Profile weight = (0 + 2)/2 = 1**
  - **Actor B's normalized ratings = -2, -1, +1**
    - **Profile weight = -2/3**

	(ActA,	ActB,	ActC,	ActD,	ActE)
Movie1(	3,	0,	0,	0,	0)
Movie2(	0,	1,	0,	0,	0,)
Movie3(	5,	0,	0,	0,	0)
Movie4(	0,	2,	0,	0,	0)
Movie5(	0,	4,	0,	0,	0)

	(ActA,	ActB,	ActC,	ActD,	ActE)
Movie1(	0,	0,	0,	0,	0)
Movie2(	0,	-2,	0,	0,	0,)
Movie3(	2,	0,	0,	0,	0)
Movie4(	0,	-1,	0,	0,	0)
Movie5(	0,	1,	0,	0,	0)

Copyright Ellis Horowitz, 2011-2022

Making Predictions

- **Given user profile *x* (movies he/she watched) and item profile *i (movies with actor profiles)***
- **Estimate the similarity of $U(x,i) = \cos(\theta) = (x.i)/(|x|\ |i|)$**

- **Technically the cosine distance is actually the angle θ and the cosine similarity is the angle 180 – θ**

- **For convenience we use cos(θ) as our similarity measure and call it the "cosine similarity" in this context**

Copyright Ellis Horowitz, 2011-2022

Pros: Content-based Approach

- No need for data on other users
- Able to recommend to users with unique tastes
- Able to recommend new and unpopular items
  - No first rater problem
- Explanations for recommended items
  - Content features that caused an item to be recommended

Copyright Ellis Horowitz, 2011-2022

Cons: Content-Based Approach

- Finding the appropriate features is not always obvious
  - E.g. movie features may be obvious but what about the features for images and music
- Overspecialization
  - Never recommends items outside user's content profile
  - People might have multiple interests
  - Unable to exploit quality judgements of other users
- Cold-start problem for new users
  - How to build a user profile

Copyright Ellis Horowitz, 2011-2022

Let's Switch Focus to Collaborative Filtering

- Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many other users (collaborating).
  - The underlying assumption of the collaborative filtering approach is that if a person *A* has the same opinion as a person *B* on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person

Collaborative Filtering

Similar Users and Jaccard Similarity

movies

4 users

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

HP:Harry Potter
SW:Star Wars
TW:Twilight

- **Consider users *x* and *y* with rating vectors *rx* and *ry***
  - ***The rating vector of user B, r_b is (5,5,4,0,0,0,0)***
- **We need a similarity metric *sim(x,y) between rating vectors***
- **The metric should capture the intuition that *sim(A,B) > sim(A,C)***
  - ***A and B both liked HP1, but A and C had very different opinions about TW and SW1***
- **Recall *sim(A,B) = |r_a intersect r_b| / |r_a union r_b |*** ***try Jaccard Similarity***
- ***sim (A,B) = 1/5; sim(A,C) = 2/4. since A&B rated only one movie in common***
  - ***Conclusion: using Jaccard Similarity we get a result we don't want, namely***
  - ***Sim(A,B) < Sim (A,C)***
- **Problem: ignores rating values**

Copyright Ellis Horowitz, 2011-2022

Cosine Similarity

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

- ***Instead of using Jaccard similarity, which ignored the rating values, let's use cosine similarity, the angle between the vectors; now we treat the unknown values as zero***
  - ***sim (A, B) = cos(r_a, r_b)***
- ***sim(A,B) = 0.38, sim(A,C) =0.32***
  - ***Now sim(A,B) > sim(A,C), which is what we wanted,*** **but not by much**
- **Problem: by treating missing ratings as zero, we sort of got the result we wanted, but actually the similarity of *A* and *C* should be farther apart than what we computed; using zero was not a great idea**

Copyright Ellis Horowitz, 2011-2022

Centered Cosine Captures User Preferences

Solution: Normalize ratings by subtracting row mean

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	2/3			5/3	−7/3		
B	1/3	1/3	−2/3				
C				−5/3	1/3	4/3	
D		0					0

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	2/3			5/3	−7/3		
B	1/3	1/3	−2/3				
C				−5/3	1/3	4/3	
D		0					0

- *The average rating for A is 10/3;*
- *The average rating for B is 14/3;*
- *The solution is to subtract the average rating for each user's score; e.g. user A and HP1 we get 4 – (10/3)= 2/3*
- *The resulting matrix*

- *sim(A,B) = cos(r_a,r_b) = 0.09; sim(A,C)=-0.56*

Result: A and C are very DISsimilar

- *sim(A,B) > sim(A,C)*

- Captures intution better
  - Missing ratings treated as average
  - Handles "tough raters" and "easy raters"

Note: Summing the rows for any user gives zero, so positive ratings means they liked the movie

Another name for centered cosine is **Pearson Correlation**

Copyright Ellis Horowitz, 2011-2022

Making Rating Predictions for a User

- Let r_x be the vector of user x's ratings
- Let N be the set of K users most similar to x who have also rated item i
- Prediction for user x and item i

- Option 1: $r_{xi} = 1/k \sum_{y \in N} (r_{yi})$ *use the average rating of users who rated i*

- Option 2: $r_{xi} = \sum_{y \in N} (s_{xy}\, r_{xy}) / \sum_{y \in N} s_{xy}$ This is a weighted average

  where $s_{xy} = sim(x,y)$

Copyright Ellis Horowitz, 2011-2022

Item-Item Collaborative Filtering

- **So far: we have used cosine similarity for user-user collaborative filtering**
- **We try to apply it to content filtering**
  - **For item *i*, find other similar items**
  - **Estimate rating for item *i* based on ratings for similar items**
  - **Use the same similarity metrics and prediction functions as in user-user model**

N(i;x) is the neighborhood of items that are rated by user x and similar to item i

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

s_{ij}… similarity of items *i* and *j*
r_{xj}…rating of user *x* on item *j*
N(i;x)… set items rated by *x* similar to *i*

Copyright Ellis Horowitz, 2011-2022

Let's Do An Example

Item – Item Collaborative Filtering (|N| =2)

users

movies	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3			5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

- unknown rating - rating between 1 to 5

List of similarities of movie 1

users

movies	1	2	3	4	5	6	7	8	9	10	11	12	sim(1,m)
1	1		3		?	5			5		4		1.00
2			5	4			4			2	1	3	-0.18
3	2	4		1	2		3		4	3	5		0.41
4		2	4		5			4			2		-0.10
5			4	3	4	2					2	5	-0.31
6	1		3		3			2			4		0.59

Here we use Pearson correlation as similarity:
1) Subtract mean rating m_i from each movie i
$m_1 = (1+3+5+5+4)/5 = 3.6$
row 1: [-2.6, 0, -0.6, 0, 0, 1.4, 0, 0, 1.4, 0, 0.4, 0]
2) Compute cosine similarities between rows

Goal: Estimate rating of movie 1 by user 5

N is 2, looking at the two nearest neighbors

Conclusion: user 5 will like movie 1: 2.6

Movie 3 and 6 are the two recommendations that are most similar to movie 1
Compute similarity weights
$S_{13} = 0.41$ $S_{16} = 0.59$
Use the weighted average to compute
$R_{15} = (0.41*2+0.59*3)/(0.41+0.59)=2.6$

Copyright Ellis Horowitz, 2011-2022

Item-Item vs. User-User

- In theory user-user and item-item are dual approaches
- In practice, item-item outperforms user-user in many use cases
- Items are "simpler" than users
  - Items belong to a small set of "genres", users have varied tastes
  - *Item* similarity is more meaningful than *user* similarity

Copyright Ellis Horowitz, 2011-2022

Collaborative filtering: Algorithm Complexity

- **Expensive step is finding *k* most similar users (or items):*O(|U|)***
  - **|*U*| = size of utility matrix**
- **Too expensive to do at runtime**
  - **Could pre-compute**
    - **The set of similar items for every item**
    - **The set of similar users for every user**
  - **Naïve pre-computation takes time *O(n*|U|)*** **This is potentially a large number**
    - **Where *n* = number of users (items)**
  - **But we can use a previous technique, e.g.**
    - **Clustering**

Copyright Ellis Horowitz, 2011-2022

Problems with Collaborative Filtering

- **Cold Start: There needs to be enough other users already in the system to find a match.**
- **Sparsity: If there are many items to be recommended, even if there are many users, the user/ratings matrix is sparse, and it is hard to find users that have rated the same items.**
- **First Rater: Cannot recommend an item that has not been previously rated.**
  - **New items**
  - **Esoteric items**
- **Popularity Bias: Cannot recommend items to someone with unique tastes.**
  - **Tends to recommend popular items.**

Copyright Ellis Horowitz, 2011-2022

Combining Content and Collaboration

- Content-based and collaborative methods have complementary strengths and weaknesses.
- Combine methods to obtain the best of both.
- Various hybrid approaches:
  - Apply both methods and combine recommendations.
  - Use collaborative data as content.
  - Use content-based predictor as another collaborator.
  - Use content-based predictor to complete collaborative data.

Copyright Ellis Horowitz, 2011-2022

Conclusions

- Recommending and personalization are important approaches to combating information overload.
- Machine Learning is an important part of systems for these tasks.
- Collaborative filtering has problems.
- Content-based methods address these problems (but have problems of their own).
- Integrating both is best.

Copyright Ellis Horowitz, 2011-2022

Evaluation Metrics for Recommendation Engines

- ***Recall*** **– the proportion of items that a user like that were recommended**
  - t_p **= number of recommended items**
  - f_n **= the remaining items**

$$\text{Recall} = \frac{tp}{tp + fn}$$

- ***Precision*** **– out of all recommended items, how many did the user like**
  - t_p **number of recommended items**
  - $t_p + f_p$ **the total items recommended**

$$\text{Precision} = \frac{tp}{tp + fp}$$

- ***Root Mean Squared Error*** **(RMSE)**
  - **Measures error in the predicted rating**

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

- ***Mean Reciprocal Rank***
  - **The larger the MRR, the better the recommendation**

$$\text{MRR} = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{1}{r(Q_i)}$$

Copyright Ellis Horowitz, 2011-2022

Web Search Engines
Image Search

Some slides come from ImageNet:A large-scale hierarchical image database
By Deng, Dong, Socher, Li-Jia Li, Kai Li and Li Fei-Fei

Copyright Ellis Horowitz 2011-2022

Search Engines Expand their Search to Images and Videos

- Image Search
  - Search primarily based on
    - tags (FlickR, FaceBook)
    - surrounding text
    - using image features
- Leading Video Search Engines
  - Google and YouTube are the leaders
  - other video search engines include:
    - Bing videos http://www.bing.com/videos/
    - Daily Motion
    - DuckDuckGo
    - Yahoo
    - Metacafe
    - Ask
    - Yandex
- https://www.searchenginejournal.com/best-video-search-engines/360822/#close

MediaMagic Video Production

Copyright Ellis Horowitz, 2011-2022

Image and Video Searching There is a Lot of It on the Web

- **Images are returned for 19% of search queries on Google**
- **Pinterest is an American image sharing and social media service designed to enable saving and discovery of information on the internet using images, and on a smaller scale, animated GIFs and videos**
- **Visual search is a growing trend**
  - **There are over 600 million visual searches on Pinterest each month**
  - **Image-based Pinterest Ads have an 8.5% conversion rate, and Pinterest is projected to clear $1 billion a year in ad revenue by 2020"**
  - **https://www.socialmediatoday.com/news/why-visual-search-will-be-one-of-the-biggest-digital-marketing-trends-of-20/545999/**

Top 100 Google searches in the U.S.

#	Keyword	Search volume
1	facebook	144,000,000
2	youtube	143,000,000
3	amazon	119,000,000
4	weather	95,300,000
5	walmart	74,400,000
6	google	70,100,000
7	wordle	61,600,000
8	gmail	59,900,000
9	target	50,400,000
10	home depot	47,000,000

Copyright Ellis Horowitz, 2011-2022

Google Advanced Image Search

text-based query criteria

image-based query criteria

Bing Image Search

Bing Initial Page with Options: Images, Videos, Maps, News, Explore, Search History

Bing Initial Images Page; Options are: Stream, Favorites, Trending

Copyright Ellis Horowitz 2011-2022

Using Google Image Search by Similarity

- *Search By Image* is available now at images.google.com or via the "Images" tab in the right-side menu on Google.com. You should see a small camera icon on the far right side of the search bar.
- There are several ways to access it:

1. Drag and drop an image on the search bar
2. Click the camera icon to upload an image from your computer
3. Paste the URL of a photo on the web into the search bar
4. Use the Chrome or Firefox extensions that add a search option to your computer's contextual menu (right click)

Copyright Ellis Horowitz, 2011-2022

Google Image Similarity Match Example

- Google will try to match an image that you drag into the search box
- The original image
- The search results

Copyright Ellis Horowitz, 2011-2022

Bing Image Similarity Match Example

Original photo

Matching photos

Some Useful Links Describing Bing Image Search

https://blogs.bing.com/search-quality-insights/May-2018/Internet-Scale-Deep-Learning-for-Bing-Image-Search

http://searchengineland.com/bing-image-search-redesigned-to-add-more-image-details-to-the-results-218276

Copyright Ellis Horowitz, 2011-2022

Search Engines Specializing in Images

- **TinEye** is a reverse image search engine
- It primarily uses image identification rather than keywords, metadata
- Upon submitting an image, TinEye creates a "unique and compact digital signature or fingerprint" of the image and matches it with other indexed images
- This procedure is able to match even heavily edited versions of the submitted image, but will not usually return similar images in the results.

List of Image Search Engine

	Reverse Search	Price
Bing Yahoo Altavista (All 3 uses Bing algorithm)	No	Free
Ask.com (Uses Google algorithm)	No	Free
Google Image	Yes Drag and Drop Tutorial	Free
Corbis	No	Buy sell Images, some are royalty free
Imnese	No	Free
TinEye	Yes	Free
GazoPa	No	For Business
PicSearch	No	Free

https://www.searchenginejournal.com/best-image-search-engines/299963/#close

Copyright Ellis Horowitz, 2011-2022

Bing on Image Duplication

- Bing Data on Image Duplication across the Web-there is a lot of it!
- 59% have at least one duplicate; many images have many more duplicates

More than 1,000 duplicates

How Search Engines do Image Indexing – First Use Tags

- Search over tags associated with images
  - Users manually add tags to images
  - Find images with tags that match the query keyword
    - FlickR, Facebook, and many other image hosting sites ask users to tag their uploaded images
- Limitations
  - Tags require human effort to create
  - Tags may be wrong, misleading, or just too general to be useful

How Search Engines do Image Indexing – Next Use Surrounding Text

- Use text associated with images for indexing
  - Search web for images, <img src=…>

  and then use:
    - Text in URL for image filename
    - Text in HTML on page
- Example: Google Image Search for "Sunset" gives
  - Sunset at Rocky Point in Australia
  - Sunset Beach, Oahu
  - Frank Smiles at Sunset
- A single keyword like "Sunset" produces a diverse set of results, no surprise

Sunset at Rocky Point

Frank Smiles at Sunset

Sunset Beach, Oahu

How Search Engines do Image Indexing - Finally Use Feature Extraction

- Feature extraction from images is far more difficult than identifying surrounding text
- Features may be low-level and not as clearly associated with the semantics of the image as a text description

Copyright Ellis Horowitz, 2011-2022

Feature Extraction
3 Types of Image Features

- Typical examined features are those related to color, texture, and shape

1. Primitive features
   - Mean color (RGB)
   - Color Histogram
     - A representation of the distribution of colors in an image
2. Semantic features
   - Color Layout, texture etc…
3. Domain specific features
   - Face recognition, fingerprint matching etc…

General features

Copyright Ellis Horowitz, 2011-2022

Feature Extraction
Color Histograms

- Histograms are collected counts of data organized into a set of predefined bins
- A color histogram measures the intensity of the color of every pixel
- To the right is a matrix containing the intensity of an image, a value between 0 and 255
- We can divide the values in the range [0, 255] into 16 bins: [0,15]U[16,31]U. . . U[240,255] and keep count of the number of pixels that fall in each bin; we get

254	143	203	176	109	229	177	220	192	9	229	142	138	64	0	63	26	8	88	82
27	68	231	75	141	107	149	210	13	239	141	35	68	242	110	208	244	0	33	86
54	42	17	215	230	254	47	41	98	180	55	253	235	47	122	208	76	110	152	100
9	186	192	71	104	193	88	171	37	233	18	147	174	1	143	211	176	188	192	68
179	20	238	192	190	132	41	248	22	134	83	133	110	254	176	238	168	234	51	204
232	25	0	163	174	129	61	30	110	189	0	173	197	163	153	43	22	87	68	118
235	35	151	185	129	81	239	170	195	94	38	21	67	101	58	37	196	149	52	154
155	242	54	0	104	109	169	47	130	254	225	156	31	161	121	15	128	35	252	205
223	114	79	129	147	6	201	68	89	107	58	44	253	84	38	1	62	5	231	218
55	188	237	188	80	101	131	241	68	133	124	151	111	28	190	4	240	76	117	145
152	155	229	76	90	217	219	105	116	77	38	49	2	9	214	161	205	116	135	33
182	94	176	199	20	149	57	223	232	113	32	45	177	15	31	179	100	119	208	81
224	118	124	172	75	29	69	180	187	195	41	44	8	170	158	101	131	31	28	112
238	83	38	7	83	69	173	183	98	237	67	227	18	218	248	237	75	192	201	146
88	195	224	207	140	22	31	118	234	34	182	116	23	47	68	242	169	152	116	248
140	37	101	230	246	145	122	64	27	58	229	1	225	143	91	100	98	90	40	195
251	4	178	139	121	95	97	174	249	162	77	115	223	186	162	82	65	252	83	196
179	180	223	230	87	182	148	76	176	19	17	4	184	176	163	102	83	81	132	206
173	137	185	242	161	161	214	49	74	238	197	37	98	102	15	217	148	8	102	168
85	9	17	222	18	210	70	21	76	241	184	216	93	93	208	102	153	212	119	47

Copyright Ellis Horowitz 2011-2022

Feature Extraction Similar Color Content

- A **correlogram** for an image is a table indexed by color pairs, where the *d-th* entry for row *(i,j)* specifies the probability of finding a pixel of color *j* at a distance *d* from a pixel of color *i* in this image. Here *d* is chosen from a set of distance values D.

- An **autocorrelogram** captures *spatial correlation between identical colors only*. This information is a subset of the correlogram and consists of rows of the form *(i,j)* only.

Feature Extraction
Color Layout

- **Need for Color Layout**
  - ***Global*** **color features give too many false positives**
- **How color layout works**
  - **Divide the whole image into sub-blocks**
  - **Extract features from each sub-block**
- **One step further**
  - **Divide into regions based on color feature concentration**
  - **This process is called *segmentation*.**

Copyright Ellis Horowitz, 2011-2022

Example: Color layout

FIGURE 2. (a) *Butterfly* color image, (b) processed color image with 30 colors, (c) pixels from image (b) belonging to color set C_i, (d) minimum bounding rectangles (MBRs) for extracted regions used to index the image collection.

**** Image adapted from Smith and Chang : Single Color Extraction and Image Query**

Copyright Elia Horowitz, 2011-2022

Example: Texture and Shape

- Texture – an innate property of all surfaces
    - e.g. clouds, trees, bricks, hair etc…
  - Refers to visual patterns of homogeneity
  - Texture is spatial arrangement of gray levels in the image
- Shape features describe the form of object boundaries and edges
- Examples:

shapes

Copyright Ellis Horowitz, 2011-2022

How Bing Does Deep Learning of Images

- Deep learning maps an image to a vector through a process called *image embedding*
- A model is first trained using deep learning techniques to map (embed) a query and an image to a high dimensional vector space in such a way that the corresponding vectors are similar if the image is semantically relevant to the query, and further apart otherwise
- Top level image matching algorithm involves 3 steps
  1. A matching stage - to select candidate images from a huge index of images
  2. Multiple ranking stages - to use computationally expensive methods to score each candidate image independently and rank all images
  3. Multiple set ranking stages - to re-rank previous candidates lists considering information from the entire candidate set, not just independent images

Bing Deep Learning in Matching

- The matching stage selects a candidate set of images from billions of images
- During matching, it is essential to obtain a result set with high recall and moderate precision.
- query, image and page text are embedded by a deep learning model into embedding vectors. We then use an efficient vector search algorithm to scan a billions-scale index to find the N-best document vectors for a query vector.
- below: the left part shows that even though the page text 'golden butterfly hd wallpapers' is not exactly the same as the query 'gold butterflies hd background', we are able to retrieve the relevant image based on query embedding, page text embedding and image embedding, which are generated from the deep learning model as shown in the right diagram.

Bing Deep Learning in Ranking Final Stage

- We now rank all the candidate images according to their relevance to the query
- Pruning of images has occurred - we reuse the previously computed embedding vectors but do more exact calculations of semantic distance between the vectors.
- For example, in the diagram below, the page contains the text 'how to maintain a healthy and happy parrot' however we are still able to recognize that the image matches the query 'a parrot rides a tricycle' because we are looking inside the image using image embedding.

ImageNet: A Large-Scale Hierarchical Image Database

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei Dept. of Computer Science, Princeton University, USA

Copyright Ellis Horowitz, 2011-2022

Recall WordNet

- We discussed WordNet in the lecture on Question/Answering;
- WordNet is a large lexical database of English.
  - Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.
- WordNet as an ontology

Copyright Ellis Horowitz, 2011-2022

Imagenet.org

- **ImageNet** is an image database organized *according to the WordNet hierarchy*, in which each node of the hierarchy is depicted by hundreds and thousands of images.
- Currently there are an average of over five hundred images per node
- 14,197,122 images covering 21,841 synsets indexed

IMAGENET

14,197,122 images, 21841 synsets indexed

Home Download Challenges About

Not logged in. Login I Signup

An Update to the ImageNet Website and Dataset

March 11, 2021

We are proud to see ImageNet's wide adoption going beyond what was originally envisioned. However, the decade-old website was burdened by growing download requests. To serve the community better, we have redesigned the website and upgraded its hardware. The new website is simpler; we removed tangential or outdated functions to focus on the core use case—enabling users to download the data, including the full ImageNet dataset and the ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

Meanwhile, the computer vision community has progressed, and so has ImageNet. The dataset was created to benchmark object recognition—at a time when it barely worked. The problem then was how to collect labeled images at a sufficiently large scale to be able to train complex models in laboratories. Today, computer vision is in real-world systems impacting people's Internet experience and daily lives. An emerging problem now is how to make sure computer vision is fair and preserves people's privacy. We are continually evolving ImageNet to address these emerging needs.

In a FAT* 2020 paper, we filtered 2,702 synsets in the "person" subtree that may cause problematic behaviors of the model. We have updated the full ImageNet data on the website to remove these synsets. The update does not affect the 1,000 categories in ILSVRC.

In a more recent paper, we investigate privacy issues in ILSVRC. 997 out of 1000 categories in ILSVRC are not people categories; nevertheless, many incidental people are in the images, whose privacy is a concern. We first annotated faces in the images and then constructed a face-blurred version of ILSVRC. Experiments show that one can use the face-blurred version for benchmarking object recognition and for transfer learning with only marginal loss of accuracy. We release our face annotations to facilitate further research on privacy-aware visual recognition.

Team members working on these new improvements: Kaiyu Yang (Princeton), Jacqueline Yau (Stanford), Li Fei-Fei (Stanford), Jia Deng (Princeton), Olga Russakovsky (Princeton).

© 2020 Stanford Vision Lab, Stanford University, Princeton University imagenet.help.desk@gmail.com Copyright infringement

https://www.image-net.org/

Copyright Ellis Horowitz 2011-2022

ImageNet is a Knowledge Ontology

- **Below is the taxonomy for a Husky dog from WordNet and the corresponding ImageNet taxonomy of images**

- S: (n) Eskimo dog, **husky** (breed of heavy-coated Arctic sled dog)
  - *direct hypernym* / ***inherited hypernym*** / *sister term*
    - S: (n) working dog (any of several breeds of usually large powerful dogs bred to work as draft animals and guard and guide dogs)
      - S: (n) dog, domestic dog, Canis familiaris (a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds) *"the dog barked all night"*
        - S: (n) canine, canid (any of various fissiped mammals with nonretractile claws and typically long muzzles)
          - S: (n) carnivore (a terrestrial or aquatic flesh-eating mammal) *"terrestrial carnivores have four or five clawed digits on each limb"*
            - S: (n) placental, placental mammal, eutherian, eutherian mammal (mammals having a placenta; all mammals except monotremes and marsupials)
              - S: (n) mammal, mammalian (any warm-blooded vertebrate having the skin more or less covered with hair; young are born alive except for the small subclass of monotremes and nourished with milk)
                - S: (n) vertebrate, craniate (animals having a bony or cartilaginous skeleton with a segmented spinal column and a large brain enclosed in a skull or cranium)
                - S: (n) chordate (any animal of the phylum Chordata having a notochord or spinal column)
                - S: (n) animal, animate being, beast, brute, creature, fauna (a living organism characterized by voluntary movement)
                - S: (n) organism, being (a living thing that has (or can develop) the ability to act or function independently)
                - S: (n) living thing, animate thing (a living (or once living) entity)
                - S: (n) whole, unit (an assemblage of parts that is regarded as a single entity) *"how big is that part compared to the whole?"; "the team is a unit"*
                - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow) *"it was full of rackets, balls and other objects"*
                - S: (n) physical entity (an entity that has physical existence)
                - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

ImageNet is a Knowledge Ontology

- S: (n) **car**, auto, automobile, machine, motorcar (a motor vehicle with four wheels; usually propelled by an internal combustion engine) *"he needs a car to get to work"*
  - *direct hyponym* / *full hyponym*
  - ***part meronym***
    - S: (n) accelerator, accelerator pedal, gas pedal, gas, throttle, gun (a pedal that controls the throttle valve) *"he stepped on the gas"*
    - S: (n) air bag (a safety restraint in an automobile; the bag inflates on collision and prevents the driver or passenger from being thrown forward)
    - S: (n) auto accessory (an accessory for an automobile)
    - S: (n) automobile engine (the engine that propels an automobile)
    - S: (n) automobile horn, car horn, motor horn, horn, hooter (a device on an automobile for making a warning noise)
    - S: (n) buffer, fender (a cushion-like device that reduces shock due to an impact)
    - S: (n) bumper (a mechanical device consisting of bars at either end of a vehicle to absorb shock and prevent serious damage)
    - S: (n) car door (the door of a car)
    - S: (n) car mirror (a mirror that the driver of a car can use)
    - S: (n) car seat (a seat in a car)
    - S: (n) car window (a window in a car)
    - S: (n) fender, wing (a barrier that surrounds the wheels of a vehicle to block splashing water or mud) *"in Britain they call a fender a wing"*
    - S: (n) first gear, first, low gear, low (the lowest forward gear ratio in the gear box of a motor vehicle; used to start a car moving)
    - S: (n) floorboard (the floor of an automobile)
    - S: (n) gasoline engine, petrol engine (an internal-combustion engine that burns gasoline; most automobiles are driven by gasoline engines)
    - S: (n) glove compartment (compartment on the dashboard of a car)
    - S: (n) grille, radiator grille (grating that admits cooling air to car's radiator)
    - S: (n) high gear, high (a forward gear with a gear ratio that gives the greatest vehicle velocity for a given engine speed)
    - S: (n) hood, bonnet, cowl, cowling (protective covering consisting of a metal part that covers the engine) *"there are powerful engines under the hoods o cowling in order to repair the plane's engine"*
    - S: (n) luggage compartment, automobile trunk, trunk (compartment in an automobile that carries luggage or shopping or tools) *"he put his golf bag in the*
    - S: (n) rear window (car window that allows vision out of the back of the car)
    - S: (n) reverse, reverse gear (the gears by which the motion of a machine can be reversed)
    - S: (n) roof (protective covering on top of a motor vehicle)
    - S: (n) running board (a narrow footboard serving as a step beneath the doors of some old cars)
    - S: (n) stabilizer bar, anti-sway bar (a rigid metal bar between the front suspensions and between the rear suspensions of cars and trucks; serves to stabilize the chassis)
    - S: (n) sunroof, sunshine-roof (an automobile roof having a sliding or raisable panel) *"'sunshine-roof' is a British term for 'sunroof'"*
    - S: (n) tail fin, tailfin, fin (one of a pair of decorations projecting above the rear fenders of an automobile)
    - S: (n) third gear, third (the third from the lowest forward ratio gear in the gear box of a motor vehicle) *"you shouldn't try to start in third gear"*
    - S: (n) window (a transparent opening in a vehicle that allow vision out of the sides or back; usually is capable of being opened)

Copyright Ellis Horowitz, 2011-2022

Some Synsets Have Very Few Images While Some Have Many

- "Mammal" subtree (1180 synsets)
  - Average # of images per synset: 10.5K

Most populated	Least populated
Humankind (118.5k)	Algeripithecus minutus (90)
Kitty, kitty-cat (69k)	Striped muishond (107)
Cattle, cows (65k)	Mylodonitid (127)
Pooch, doggie (62k)	Greater pichiciego (128)
Cougar, puma (57k)	Damaraland mole rat (188)
Frog, toad (53k)	Western pipistrel (196)
Hack, jade, nag (50k)	Muishond (215)

Copyright Ellis Horowitz, 2011-2022

Constructing ImageNet

- 2-step process

Step 1 :
Collect candidate images Via the Internet

Step 2 :
Clean up the candidate Images by humans

Copyright Ellis Horowitz, 2011-2022

Step 1:Collect Candidate Images from the Internet

- For each synset, the queries are the set of WordNet synonyms
- Accuracy of Internet Image search results: 10 %
  - For 500-1000 clean images, needs 10K images
- Query expansion
  - Synonyms: German police dog, German shepherd dog
  - Appending words form ancestors: sheepdog, dog
- Multiple Languages
  - Italian, Dutch, Spanish, Chinese e.g. 德国牧羊犬，pastore tedesco
- More engines: Yahoo! , flickr, Google
- Parallel downloading

Copyright Ellis Horowitz, 2011-2022

Step 2:Clean Up the Candidate Images by Humans

- Rely on humans to verify each candidate image collected for a given synset
- Amazon Mechanical Turk (AMT)
  - used for labeling vision data
  - 300 images: 0.02 dollar
  - 14,197,122 images: 946 dollars
  - 10 repetition: 9460 dollars
  - Jul 2008 -Apr 2010:11 million images
- Present the users with a set of candidate images and the definition of the target synset
- let users select the best match ones

amazonmechanical turk
Artificial Artificial Intelligence

Copyright Ellis Horowitz, 2011-2022

A Task on AMT

(amazon mechanical turk)

Workers do annotation on AMT
-Multiple annotations for each images

Annotation Results
- An average of > 97% accuracy

Copyright Ellis Horowitz, 2011-2022

Ensure Accuracy

- Users Enhancement
- Provide wiki and Google links for definitions
- Make sure workers read the definition
  - Definition quiz
- Allow more feedback. E.g. “unimaginable synset” expert opinion

Copyright Ellis Horowitz, 2011-2022

Ensure Accuracy

- Human users make mistakes
- Not all users follow the instructions
- Users do not always agree with each other
  - Subtle or confusing synsets, e.g. Burmese cat
- Quality Control System

User 1	Y	Y	Y
User 2	N	Y	Y
User 3	N	Y	Y
User 4	Y	N	Y
User 5	Y	Y	Y
User 6	N	N	Y

#Y	#N	Conf Cat	Conf BCat
0	1	0.07	0.23
1	0	0.85	0.69
1	1	0.46	0.49
2	0	0.97	0.83
0	2	0.02	0.12
3	0	0.99	0.90
2	1	0.85	0.68

Copyright Ellis Horowitz, 2011-2022

Quality Control System

- randomly sample an initial subset of images to users
  - Have multiple users independently label same image
- obtain a confidence score table, indicating the probability of an image being a good image given the user votes
  - Different categories requires different levels of consensus
- Proceed until a pre-determined confidence score threshold reached

Copyright Ellis Horowitz, 2011-2022

Properties of ImageNet

- **Scale**: 12 subtrees, 3.2 million images, 5247 categories

- **Hierarchy**: densely populated semantic hierarchy, based on WordNet

Summary of selected subtrees

Subtree	# Synsets	Avg. synset size	Total # image
Mammal	1170	737	862K
Vehicle	520	610	317K
GeoForm	176	436	77K
Furniture	197	797	157K
Bird	872	809	705K
MusicInstr	164	672	110K

Copyright Ellis Horowitz, 2011-2022

Properties of ImageNet

- **Accuracy: clean dataset at all level**

- Diversity: variable appearances, positions, view points, poses, background clutter, occlusions.

Copyright Ellis Horowitz, 2011-2022

ImageNet Applications

- ImageNet is useful for many computer vision applications such as object recognition, image classification and object localization.
- Images for ImageNet were collected from various online sources. ImageNet doesn't own the copyright for any of the images.
  - For public access, ImageNet provides image thumbnails and URLs from where the original images were downloaded. Researchers can use these URLs to download the original images.
  - Those who wish to use the images for non-commercial or educational purpose, can create an account on ImageNet and request access. This will allow direct download of images from ImageNet. This is useful when the original sources of images are no longer available.
- Non-parametric Object Recognition
- Tree Based Image Classification
- Automatic Object Localization

Copyright Ellis Horowitz, 2011-2022

Pros and Cons

Pros

1. Crowdsourcing
2. Benchmarking
3. Open: Download Original Images, URLs, Features, Object Attributes, API

Cons

1. Improve algorithm: PageRank
2. AMT: hierarchical users based on their ability
3. Only one tag per image

Copyright Ellis Horowitz, 2011-2022

Summary by the ImageNet Developers

- **ImageNet is intended to serve as**
  - A dataset
  - A knowledge ontology
- **Construction of large-scale image dataset is a relatively new research area**
  - Crowdsourcing might be the future of many such tasks
  - see an ImageNet developer's TED talk, https://www.youtube.com/watch?v=40riCqvRoMs
- **Benchmarking: what does classifying 10k+ image categories tell us?**
  - Computation matters
  - Size matters
  - Density matters
  - Hierarchy matters

Copyright Ellis Horowitz, 2011-2022