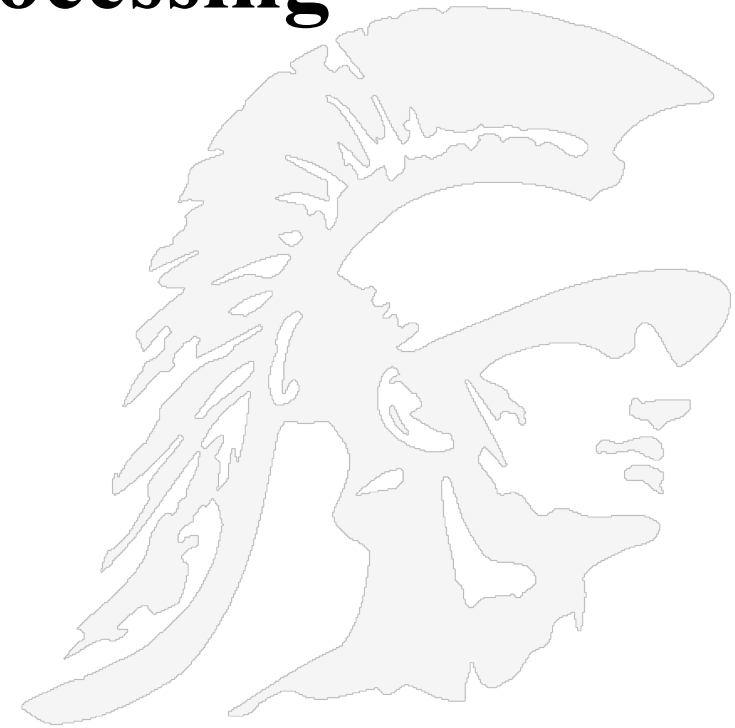
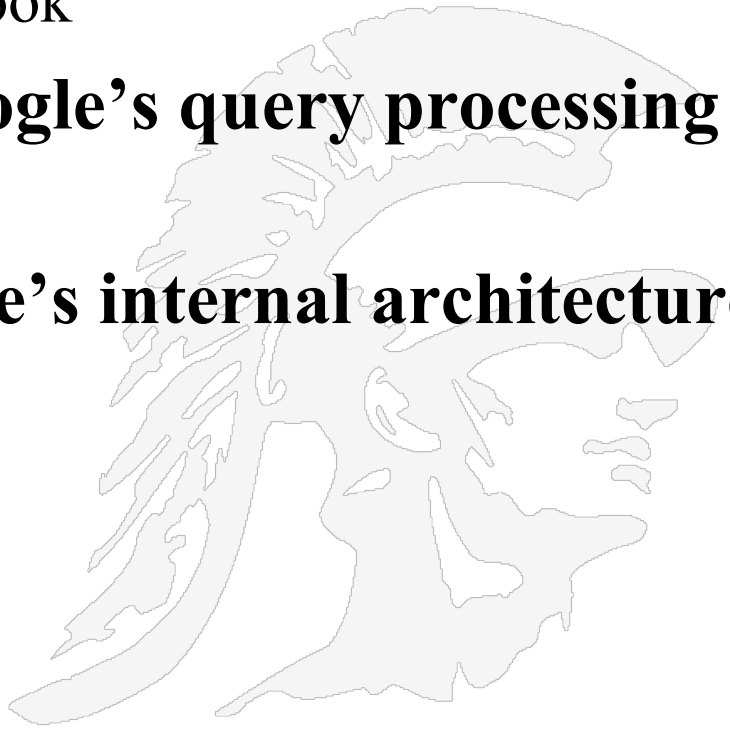
Query Processing



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**



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 and profile to create a thorough response
- **The following slides contain heuristics that can be applied to speed up step 1 of the process**

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 don't alter rank-ordering much**
- **Benefit:**
  - Postings of low-idf terms have many docs → these (many) docs will eventually get eliminated from the set of contenders, so it is best to avoid them initially

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

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 champion list for *t*
  - (aka fancy list or top docs 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 amongst these

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

Strategy 4: Introduce an Authority Measure

- **Assign to each document a *query-independent* quality score in [0,1] to each document *d***
  - **Denote this by *g(d), g stands for goodness***
- **Thus, a quantity like the number of citations is scaled into the range of [0,1]**
- **The PageRank can also be scaled to the same range**
- **Heavily curated content, e.g. Wikipedia can be given a high quality score**

Net Score

- **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 relevance and authority**
  - **We could use some other linear combination, assigning different weights to the two factors; for now we use weights = *1***
  - **In processing a query the algorithm now seeks the top K docs by net score**

Strategy 5: Reorganize the Inverted List

- So far we assumed that all documents were ordered by docID
- 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

Champion lists in *g(d)*-ordering

- Can 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

cosine score

Strategy 6: High and Low Lists Heuristic

- **For each term, we 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 tiers**

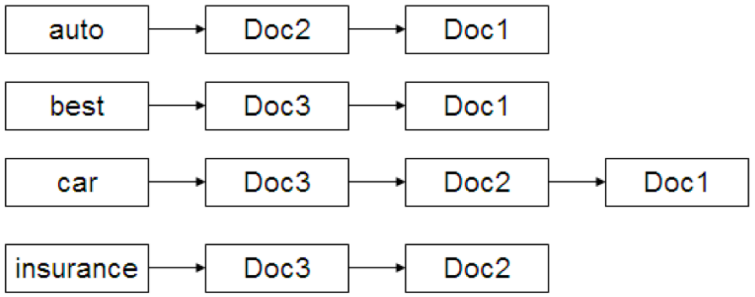
Example of a Tiered Inverted Index

	Doc1	Doc2	Doc3
car	27	4	24
auto	3	33	0
insurance	0	33	29
best	14	0	17

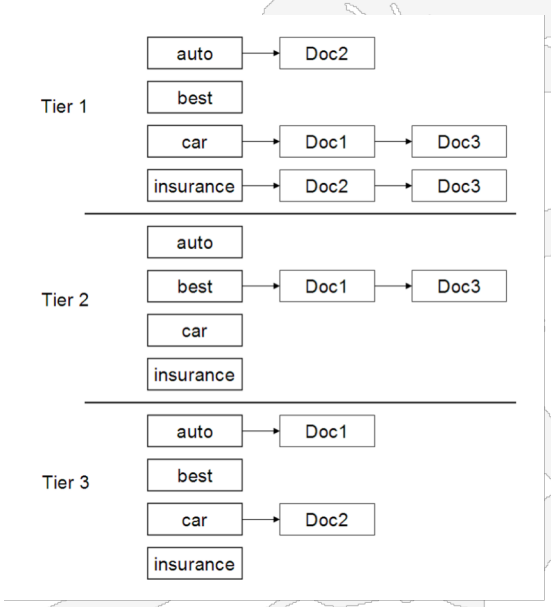
4 documents with term frequencies

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

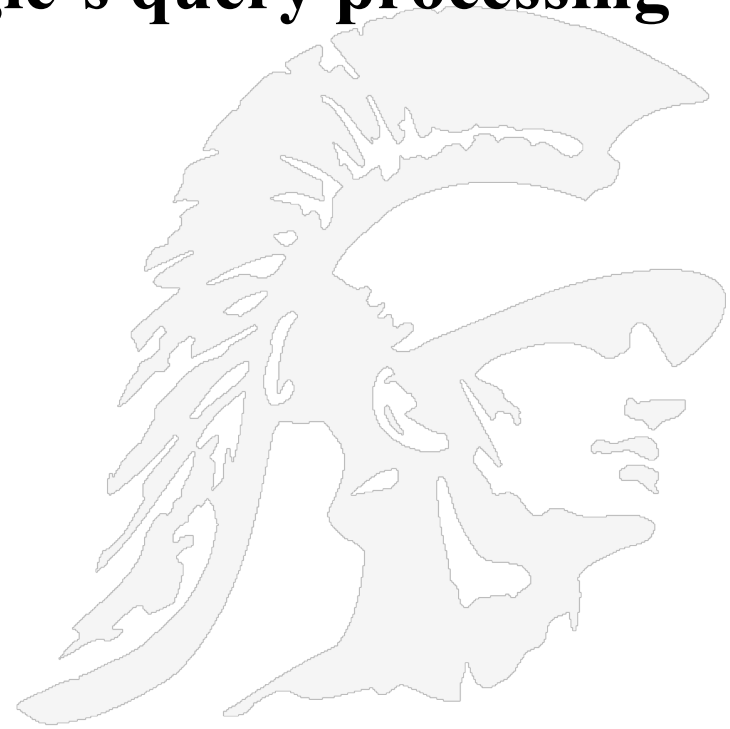


Static quality-ordered index;
Assume doc1, doc2, doc3 have quality scores *g(1)=0.25, g(2)=0.5, g(3)=1*



Tiered index, if tier 1 doesn't provide enough results, try tier 2, etc

- **Now lets switch gears and look at the problem of reverse engineering Google's query processing algorithm**



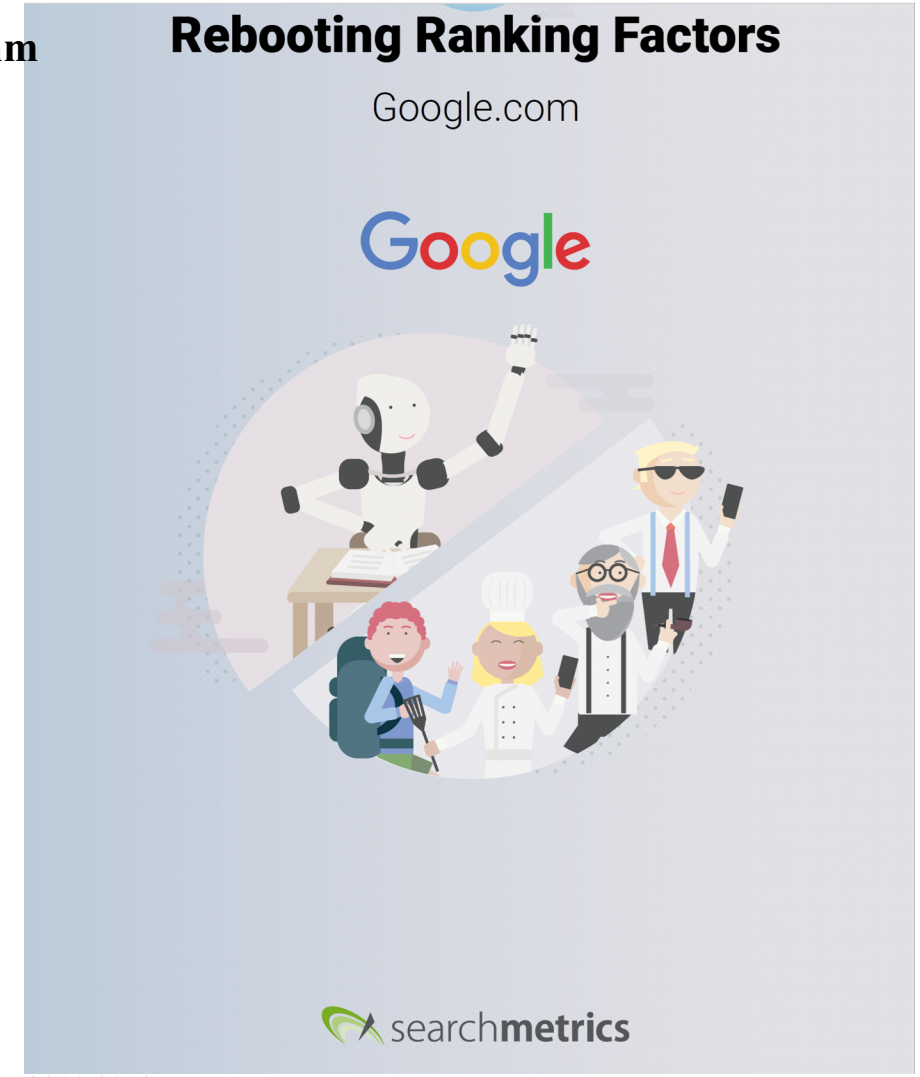
SEO Companies Carefully Track Ranking Factors Valued by Google

Reverse engineering the Google ranking algorithm

Download at:
http://www-scf.usc.edu/~csci572/papers/Searchmetrics.pdf

Ranking factors are organized into categories:

1. **Content factors**
   - Content relevance
   - Word count
2. **User signals**
   - Click Through Rate (CTR)
   - Bounce rate
3. **Technical factors**
   - Presence of H1/H2
   - Use of HTTPS
4. **User experience**
   - Number of internal/external links
5. **Social signals**
   - Facebook total
   - Tweets

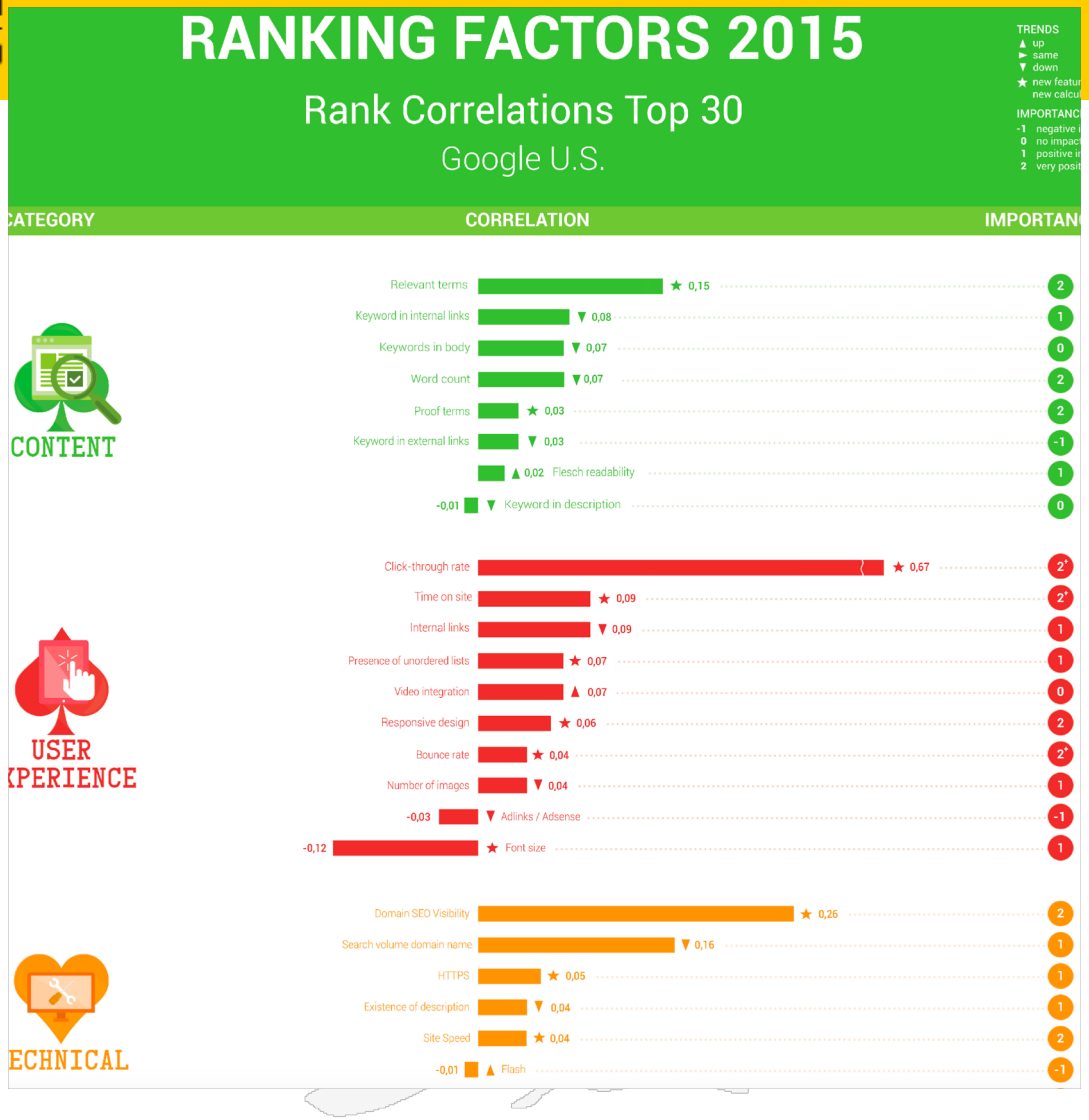


Here are the Top Two Factors in Each Category

Relevant terms
Keyword in internal links

Click Through Rate
Time on site

Domain visibility?
.com sites are favored

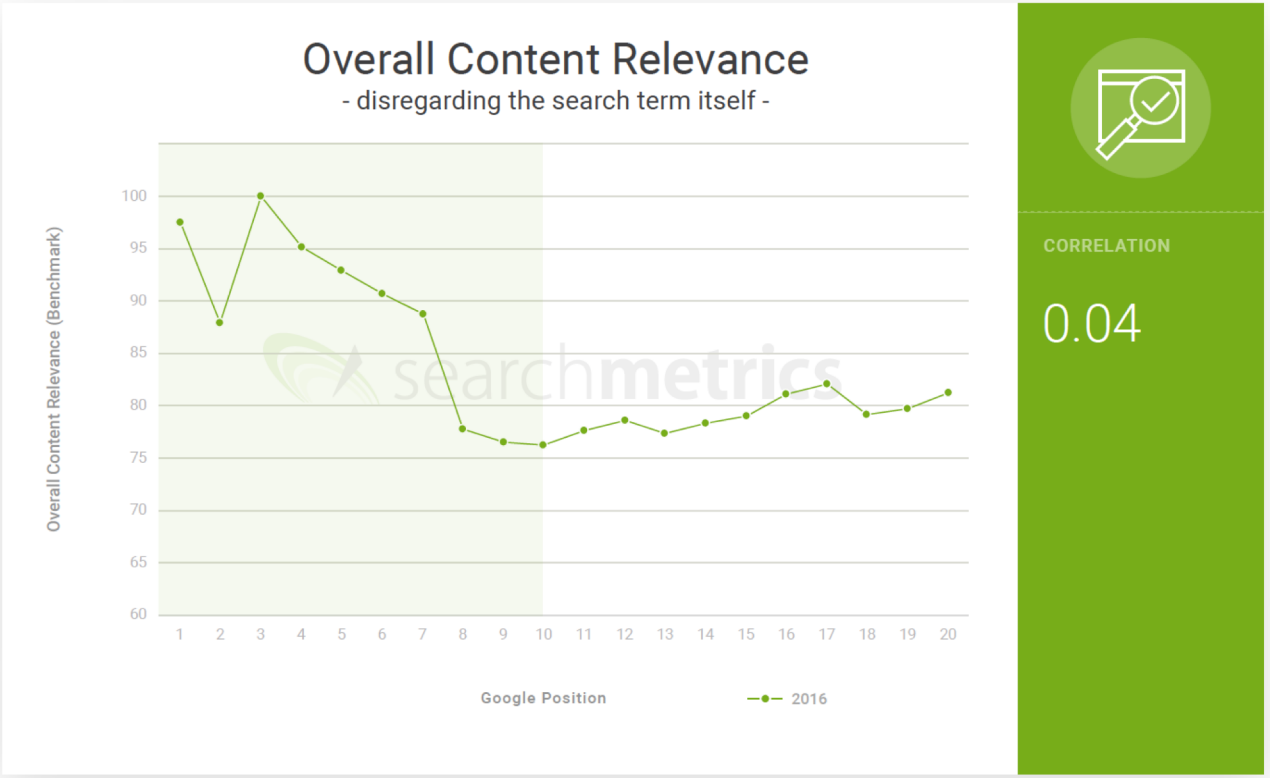


Content Factors

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.

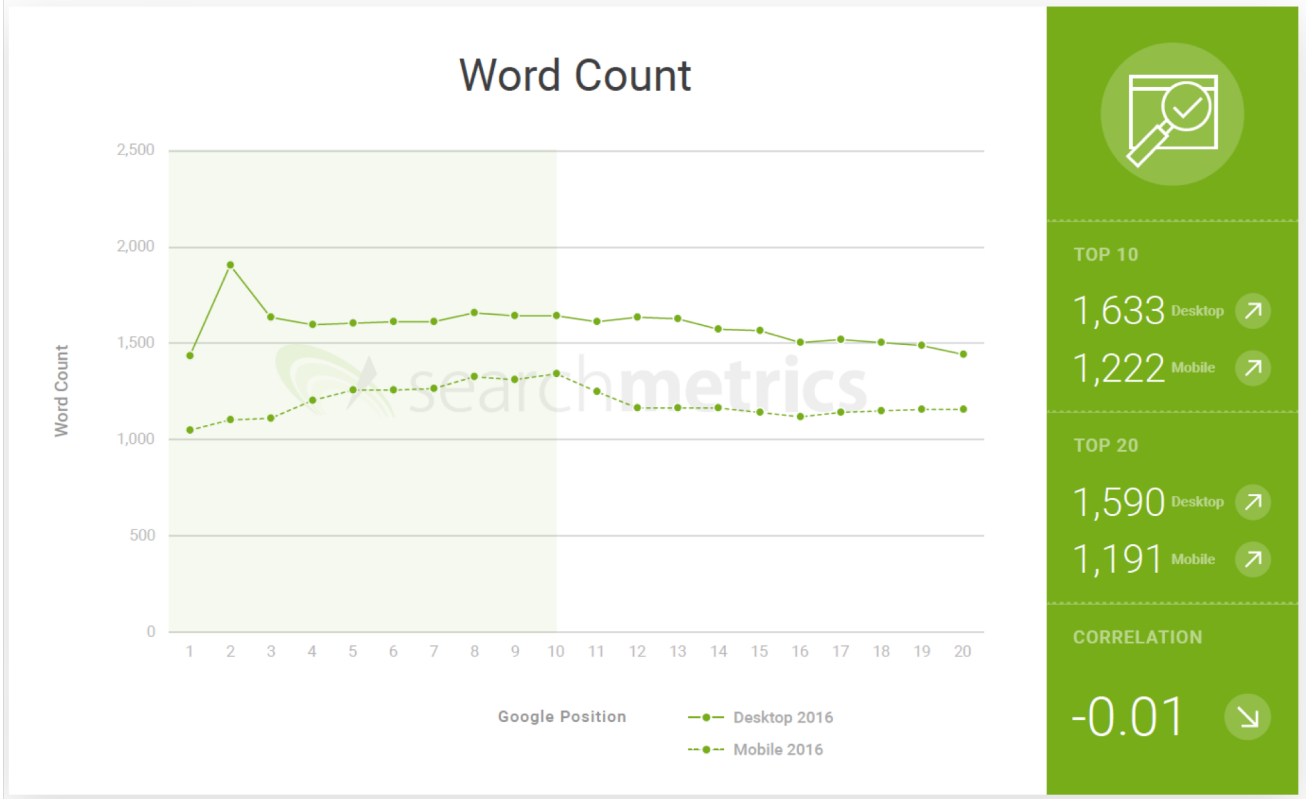
Thereafter, the landing pages on subsequent positions show lower relevance scores



Word Count

- The word count of a landing page ranked among the top positions and has been rising for years

- Pages rank well under the condition that the content is not simply long, but also relevant,

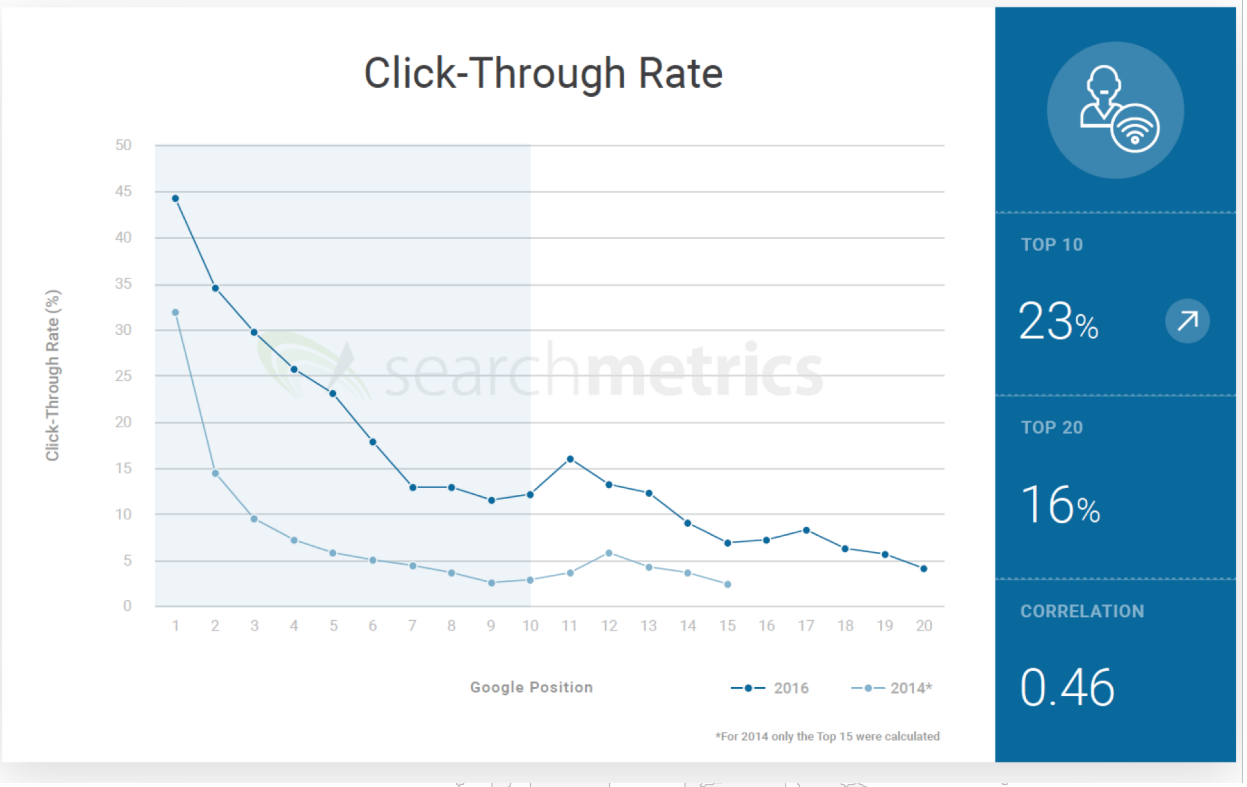


Click-Through Rate

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.

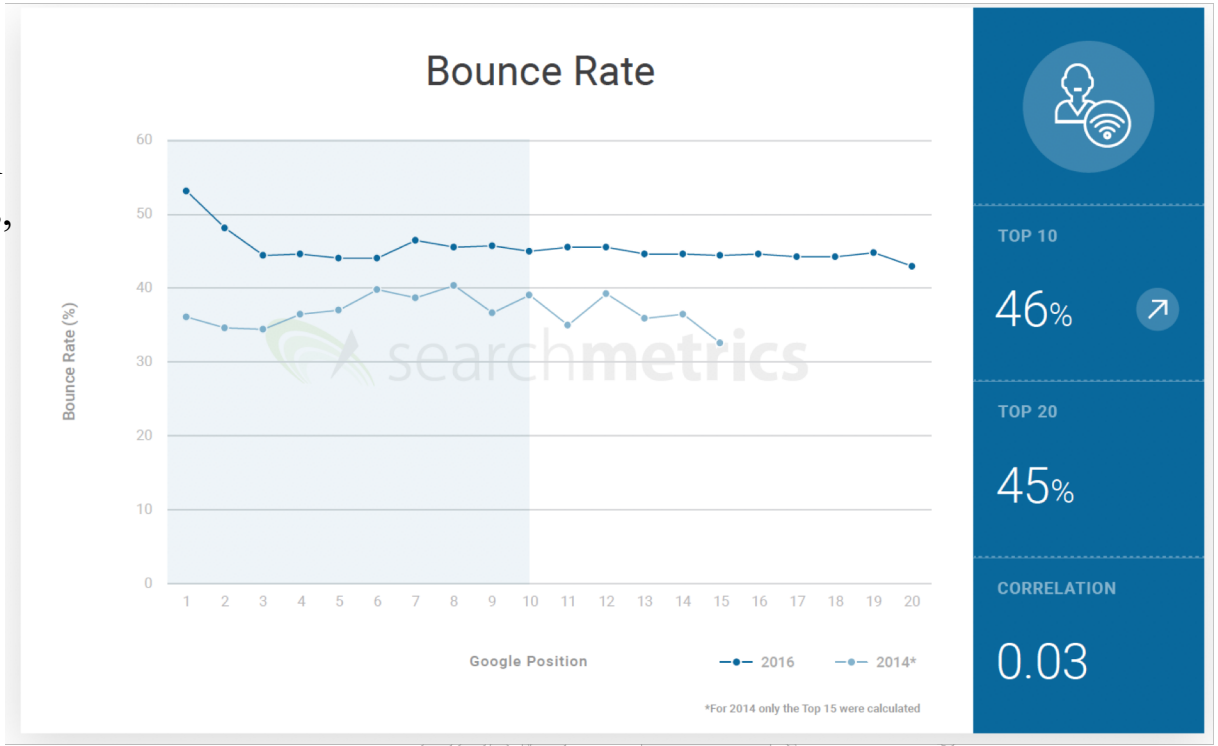


*SERP: Search Engine Results Page

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*.

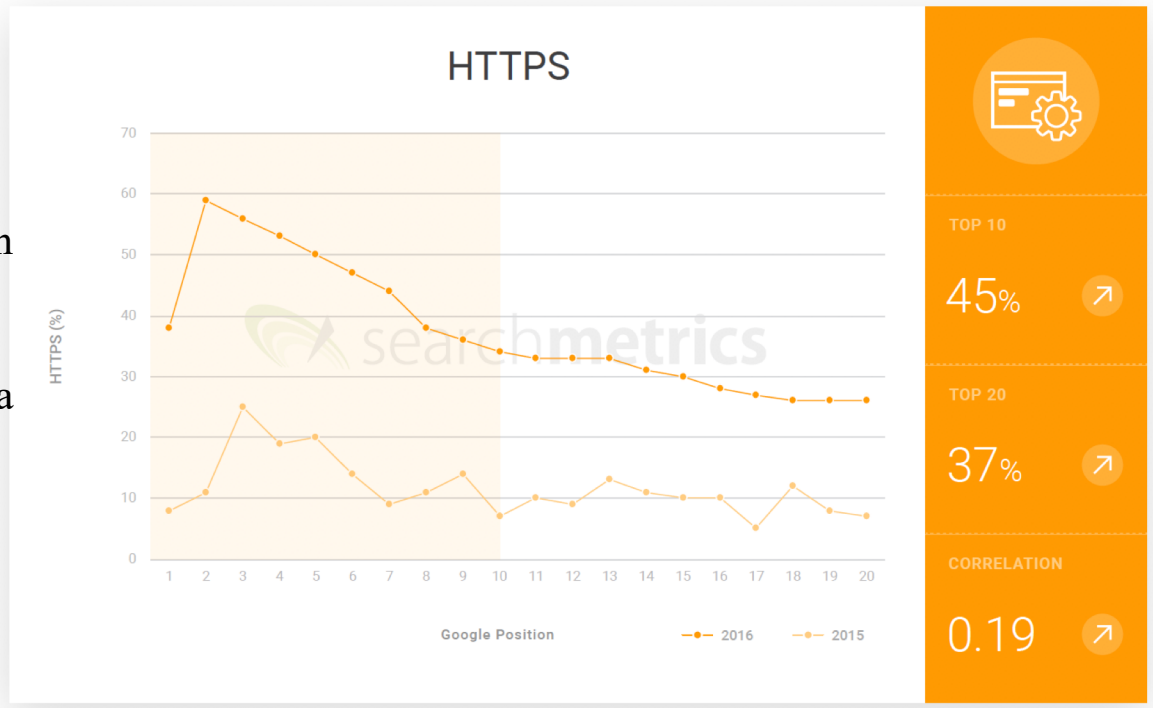
These are single-page sessions where the user leaves the site without interacting with the page.



*SERP: Search Engine Result Page

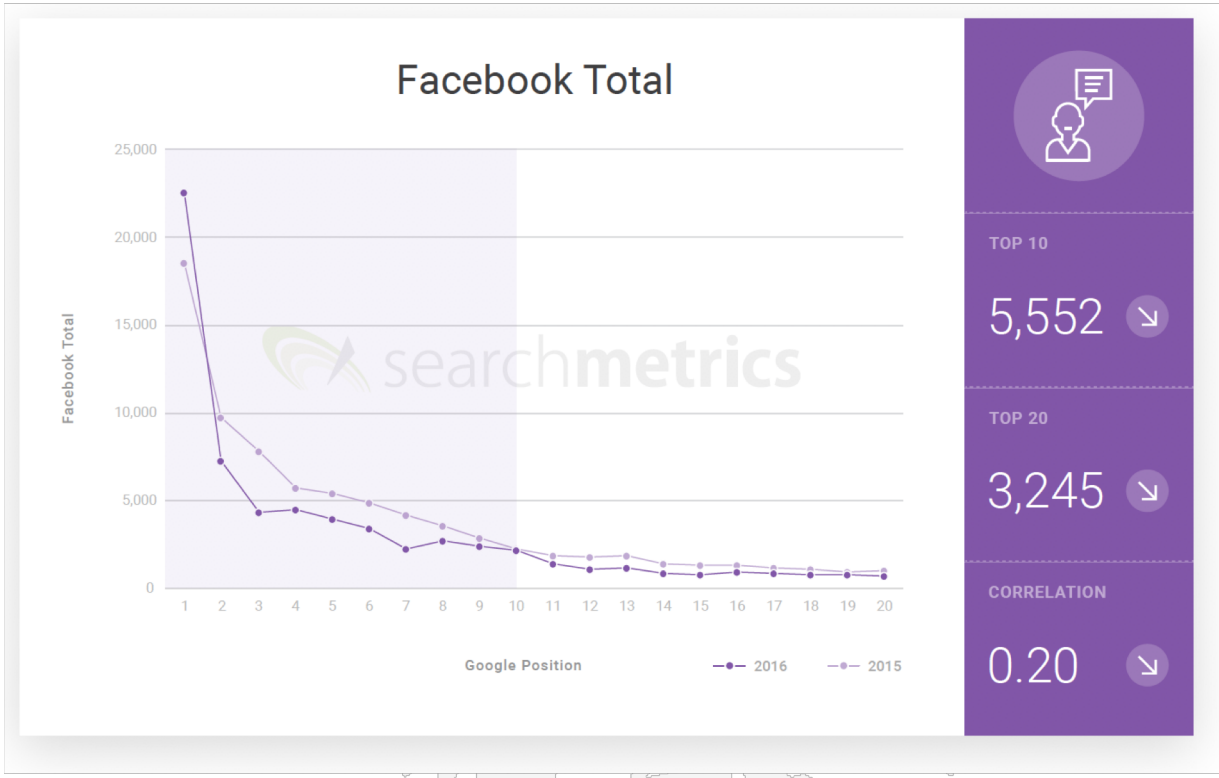
Technical Factors

- 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.



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



Mobile Friendliness

All top 100 websites have a mobile-friendly version; they use either a mobile sub-domain or responsive design;

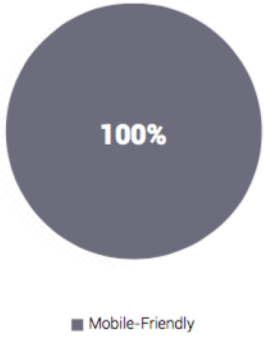
Separate mobile websites are diminishing in popularity, but e.g. try Sephora.com on desktop and mobile. Last time I looked they were still using m.sephora.com

Over a fifth of websites outside of the top 100 offer no mobile-friendly solution

Mobile-friendly websites

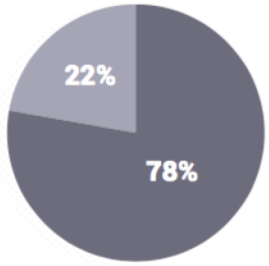
The following graphics show the frequency of websites with mobile-friendly solutions amongst the top 100 domains by SEO visibility.

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.

Finally Correlation is Not Necessarily Causation

- **See the full Searchmetrics report at**
  - http://www-scf.usc.edu/~csci572/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**

The Google Architecture

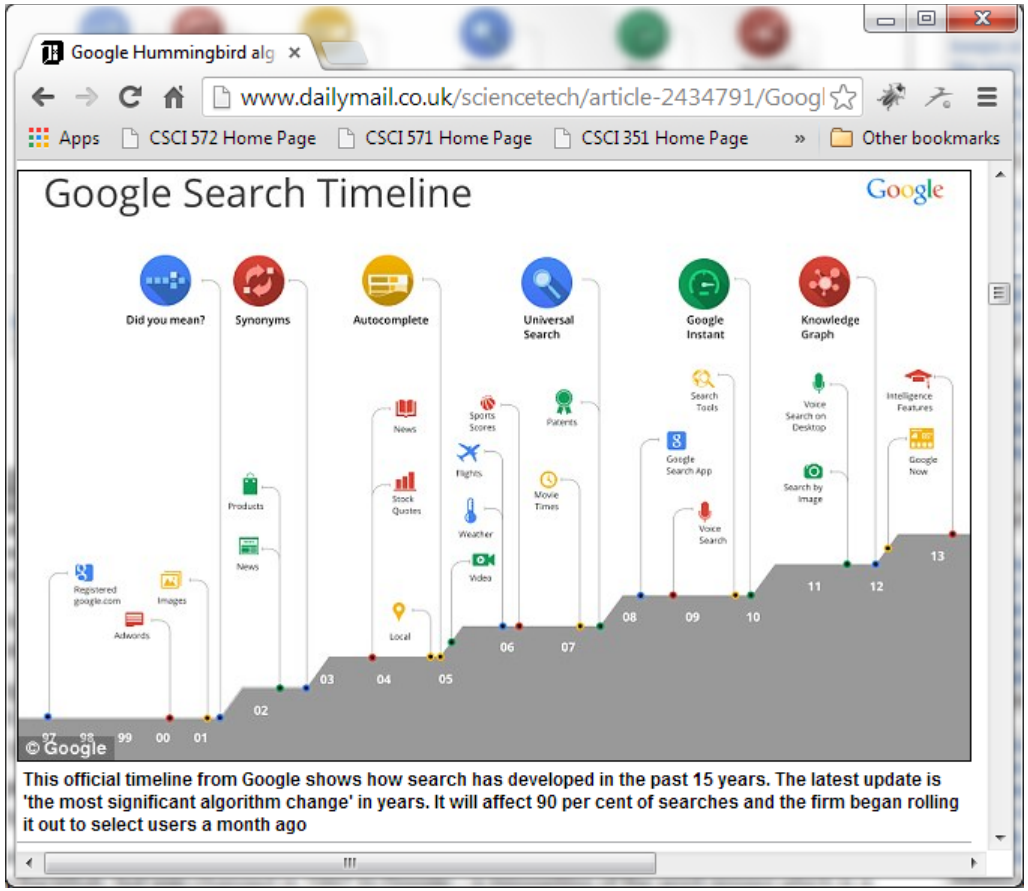
See Google's Website on how search works at

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

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



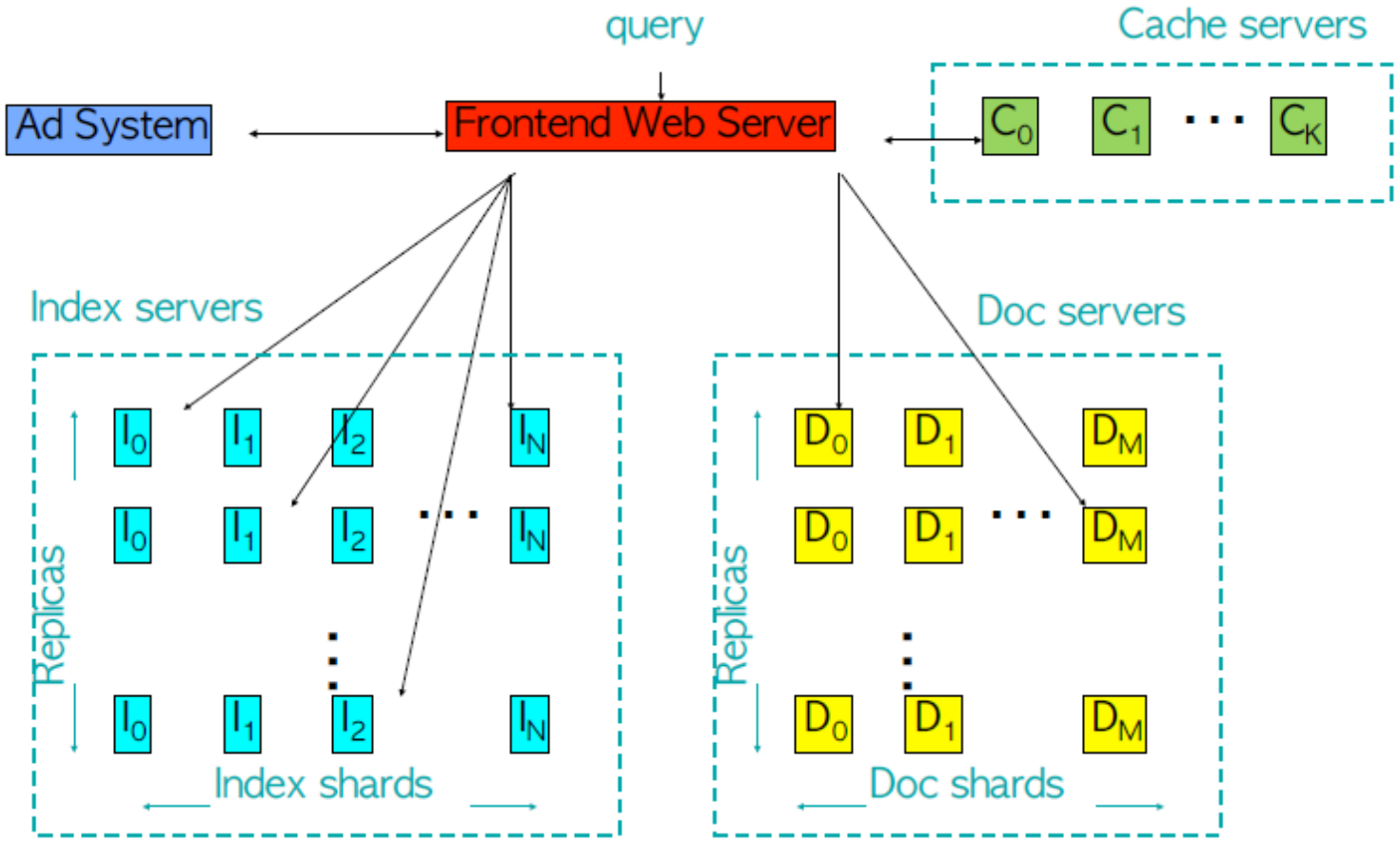
IR Challenges for Google from 1999 to 2014

- 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

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.

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

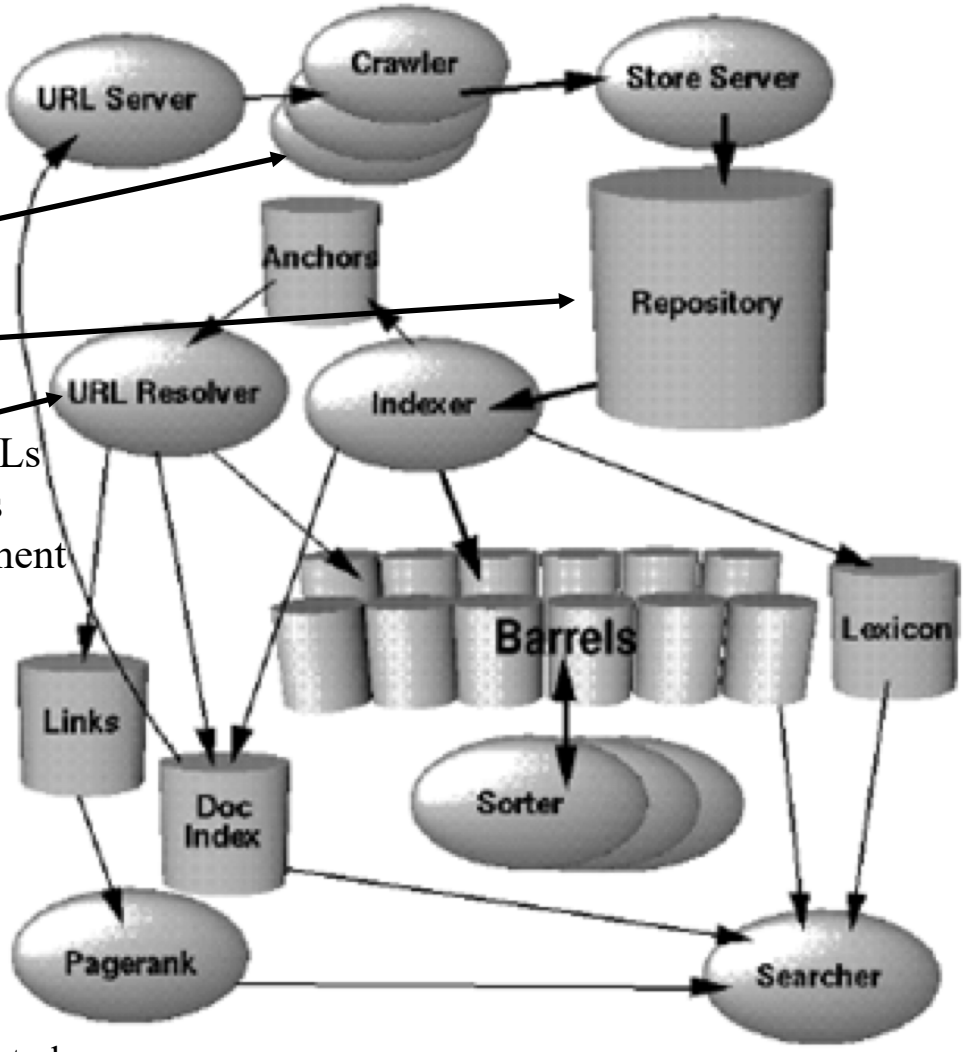


Figure 1. High Level Google Architecture

Diagram from
"*The Anatomy of a Large-Scale Hypertextual Web Search Engine*"
http://infolab.stanford.edu/~backrub/google.html

Google 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.

Google Architecture Today

- **Google data centers** combine large amounts of digital storage (mainly hard drives and SSDs), compute nodes organized in aisles of racks, internal and external networking, environmental controls and operations software (especially as concerns load balancing and fault tolerance).
- Google data centers estimated in a July 2016 report that Google at the time had 2.5 million servers.
- As of 2014, Google used a heavily customized version of Debian (GNU/Linux).
- Google ranks as the third largest ISP
- Google data Centers in the U.S.

United States:

1. Berkeley County, South Carolina
2. Council Bluffs, Iowa
3. Douglas County, Georgia
4. Jackson County, Alabama[3]
5. Lenoir, North Carolina
6. Montgomery County, Tennessee
7. Pryor Creek, Oklahoma at MidAmerica Industrial Park
8. The Dalles, Oregon

The Dalles data center is a $600 million complex built in 2006 and is approximately the size of two American football fields, with cooling towers four stories high.
The site was chosen to take advantage of inexpensive hydroelectric power, and to tap into the region's large surplus of fiber optic cable

Other Useful Videos

- **Matt Cutts 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)
  - How Search Works
    - **http://www.youtube.com/watch?v=BNHR6IQJGZs** (3 min)
- **Larry Page on the future directions of Google**
  - **http://www.youtube.com/watch?v=mArrNRWQEso** (3/2014, 23 min)

Google is More Than Search

- **Google has seven services claiming over a billion monthly active users**
  - Google Search, Google Maps, YouTube, Android, Gmail, the Play Store and Google Chrome
- **there are now over 2 billion Chrome browsers in active use**
  - https://techcrunch.com/2016/11/10/google-says-there-are-now-2-billion-active-chrome-installs/
  - Chrome had a 50.6% browser market share across PCs, mobile devices and consoles.
  - Apple Safari only has a 13.7% market share but is the second-most popular browser
  - Naturally Google search is the default on Chrome; Yahoo used to be the default search engine for Firefox, but now Google is back, and Bing is the default for Microsoft's Edge and Internet Explorer