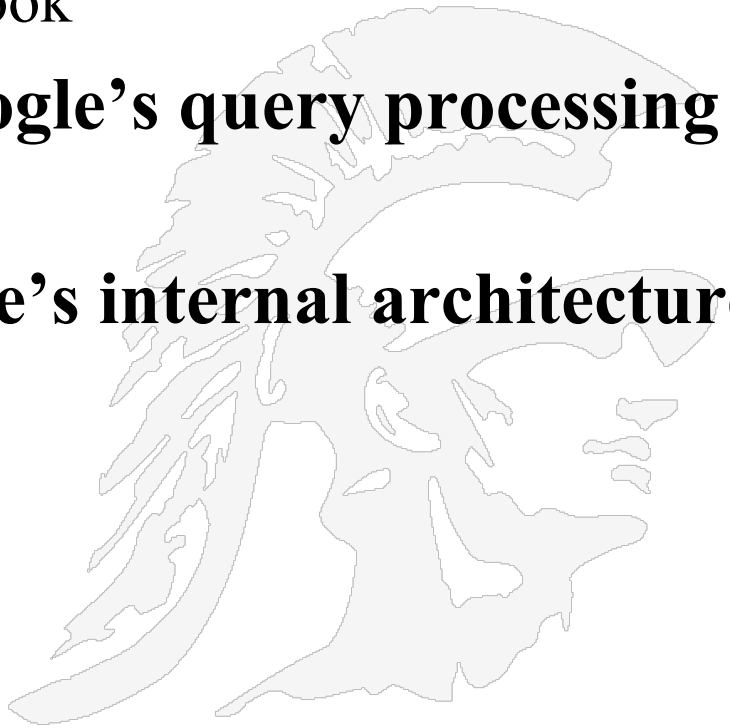
Query Processing

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, 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**

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, but 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

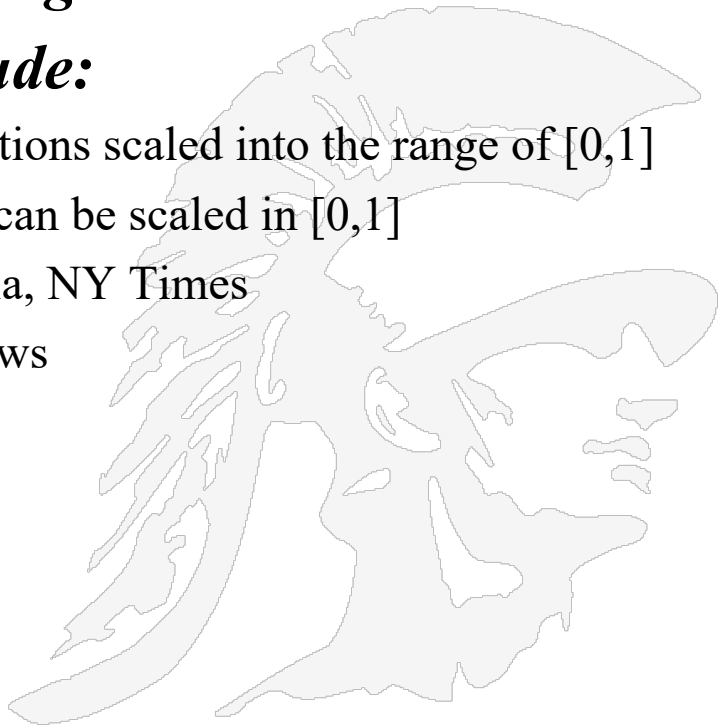
- **Pre-compute for each dictionary term *t*, the *r* docs of highest weight (tf-idf) in *t*'s postings**
  - Call this the champion list for *t*
  - (aka fancy list or top docs for *t*)
- **Note that *r* has to be chosen at index build time**
  - Thus, it's possible that *r* < *K (number of docs to 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 among 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 *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

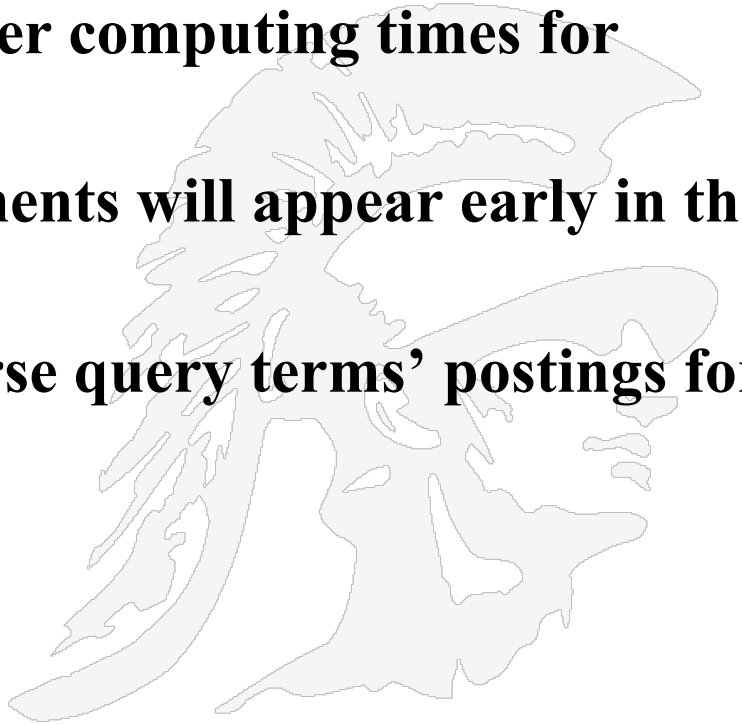


Combine Relevance and Authority

- 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 net-score

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



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

$$\text{net-score}(q,d) = g(d) + \frac{\vec{V}(q) \cdot \vec{V}(d)}{|\vec{V}(q)||\vec{V}(d)|}.$$

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 tiers**

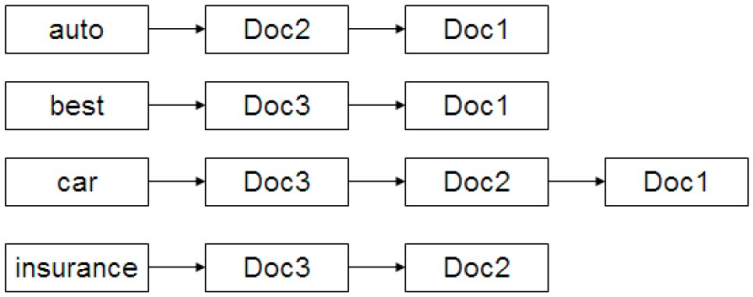
Example of a Tiered Inverted Index; A Generalization of Champion Lists

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

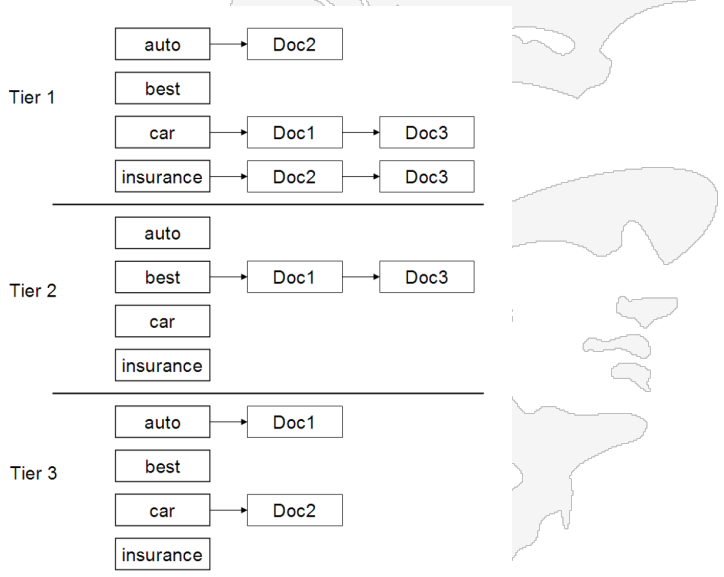
4 documents with term frequencies

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

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

Part 2: Google's Query Processing Algorithm

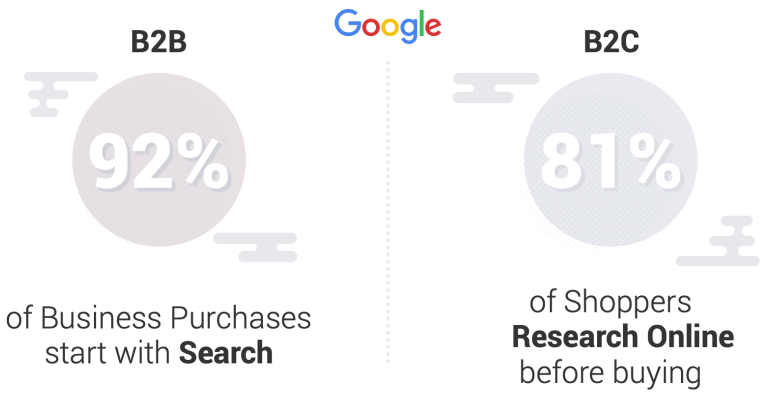
- Now let's switch gears and look at the problem of reverse engineering Google's query processing (ranking) algorithm
- There are two main companies trying to do this:
  1. *Searchmetrics* which tracks the results from thousands of keywords while analyzing the content on each page producing a ranking 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

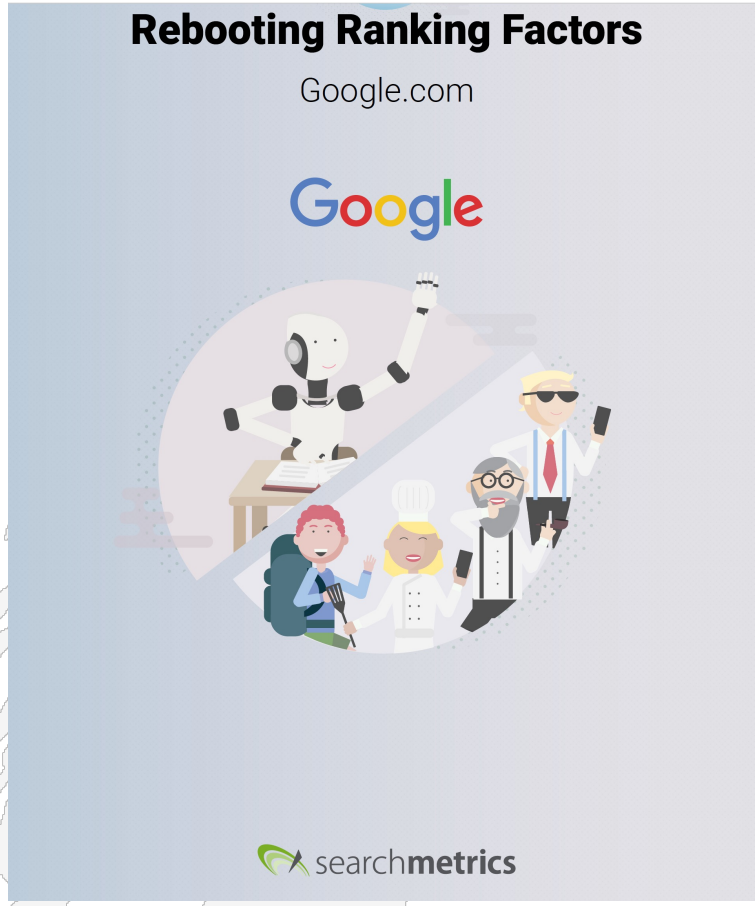
Searchmetrics Tracks Ranking Factors Valued by Google's Algorithm

Reverse engineering the Google ranking algorithm

Download at:
http://csci572.com/papers/Searchmetrics.pdf

Why search is important commercially





General Ranking Factors

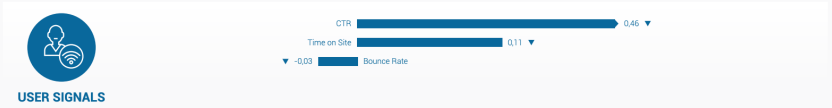
Here are the major categories and some factors

Content

Overall relevance
Word count

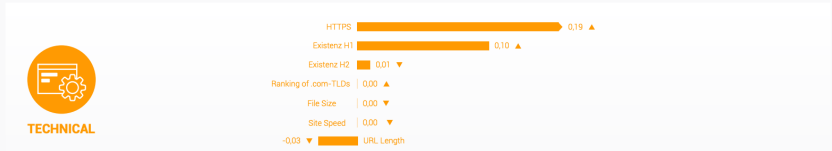
User Signals

CTR
Bounce rate



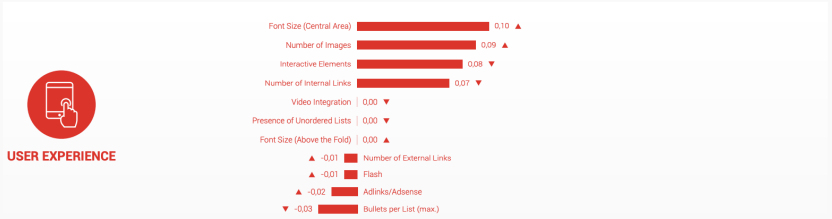
Technical

HTTPS
Presence of H1/H2



User Experience

Font size
number of internal/external links

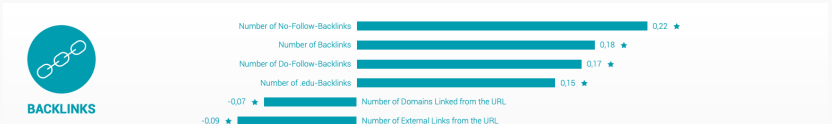


Social Signals

Pinterest
Facebook/Tweet

Backlinks

number of No-follow



Note: What Are Nofollow Links

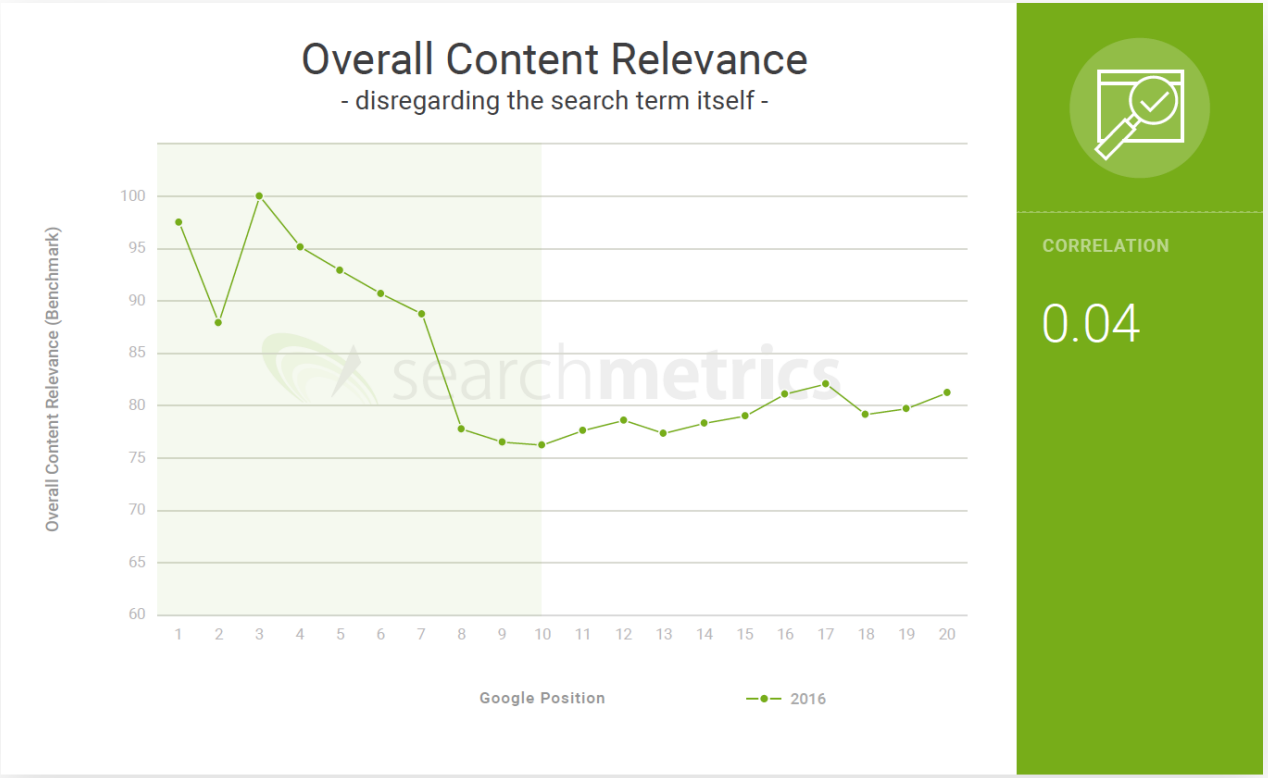
- The nofollow tag tells search engines to ignore that link
- Google introduced the rel="nofollow" option in 2005 for bloggers that were struggling with people using comment spam to try and build links in the hope of ranking for specific keywords
- Examples
- `<a href="https://example.com" rel="nofollow">Anchor 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">;`

Content Factors

The analysis shows that the content relevance, decreases as the position in the search results drops.

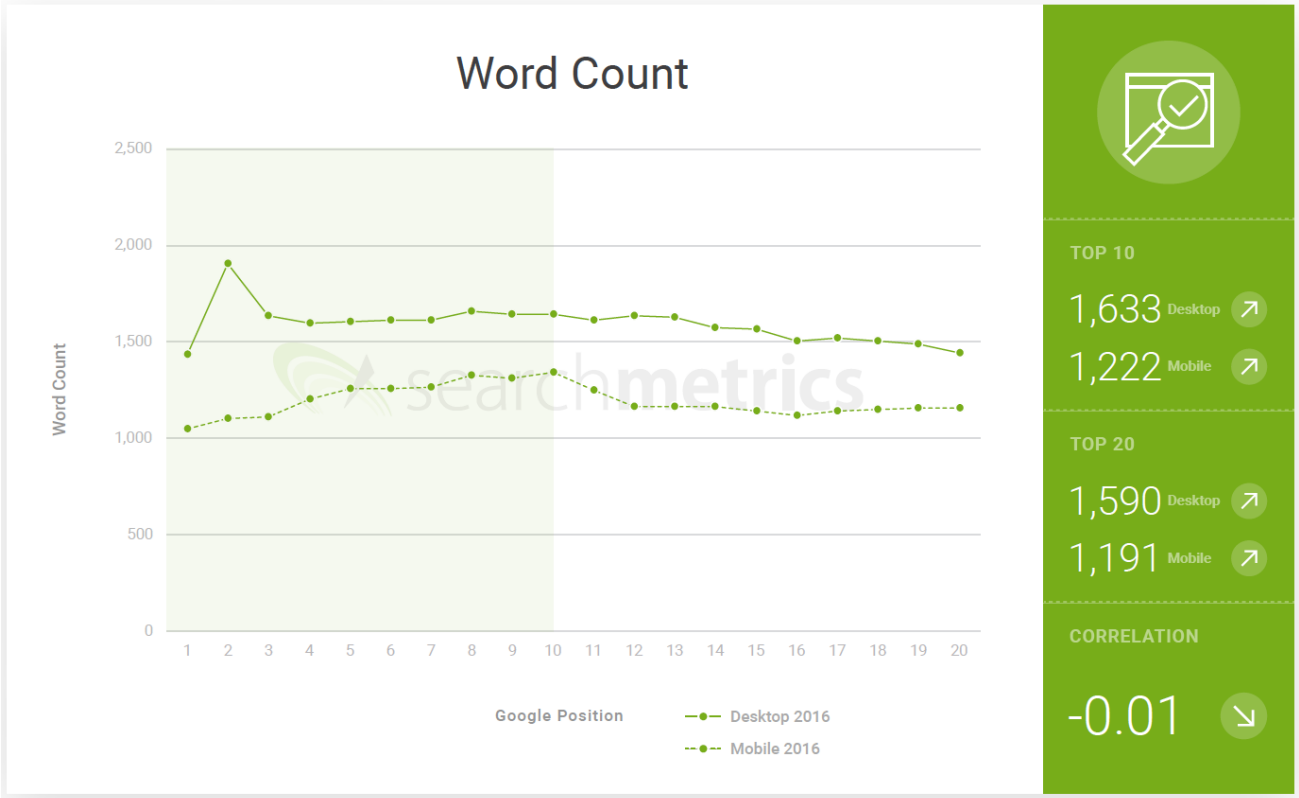
The highest content relevance scores were found among the results for positions 3 to 6.

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

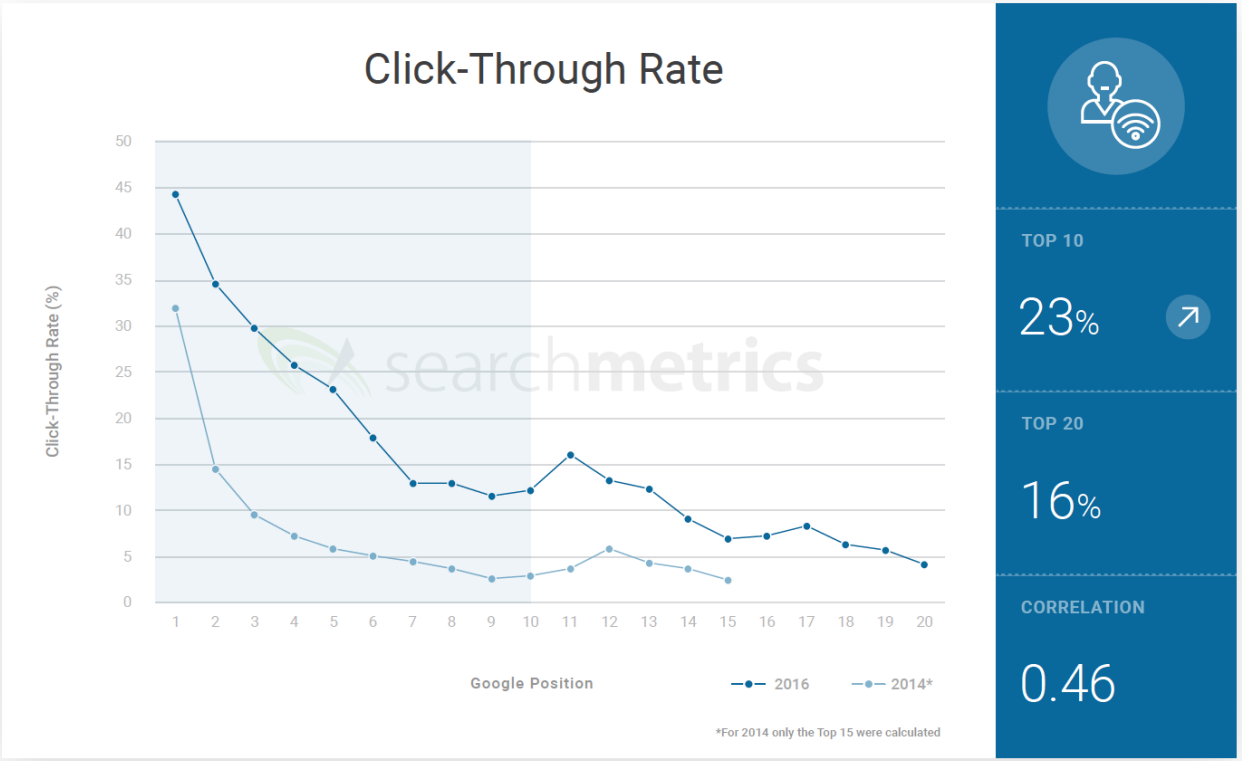


Click-Through Rate

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.

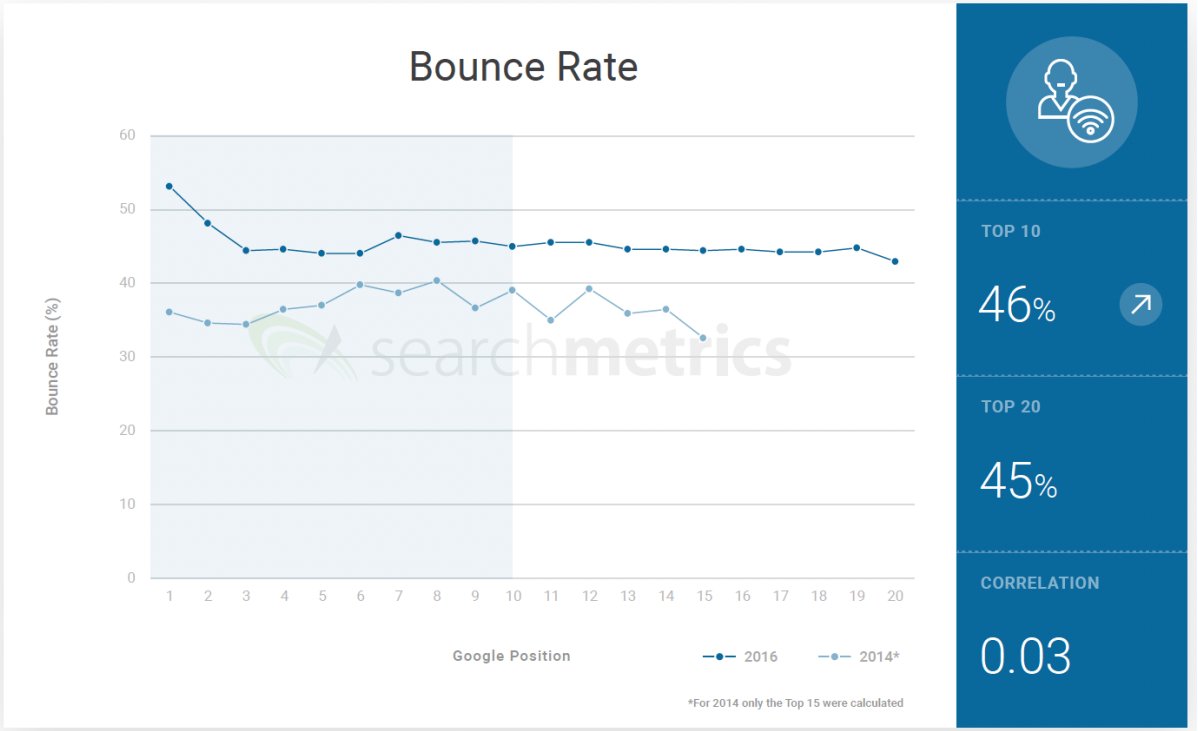


*SERP: Search Engine Results Page

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*.

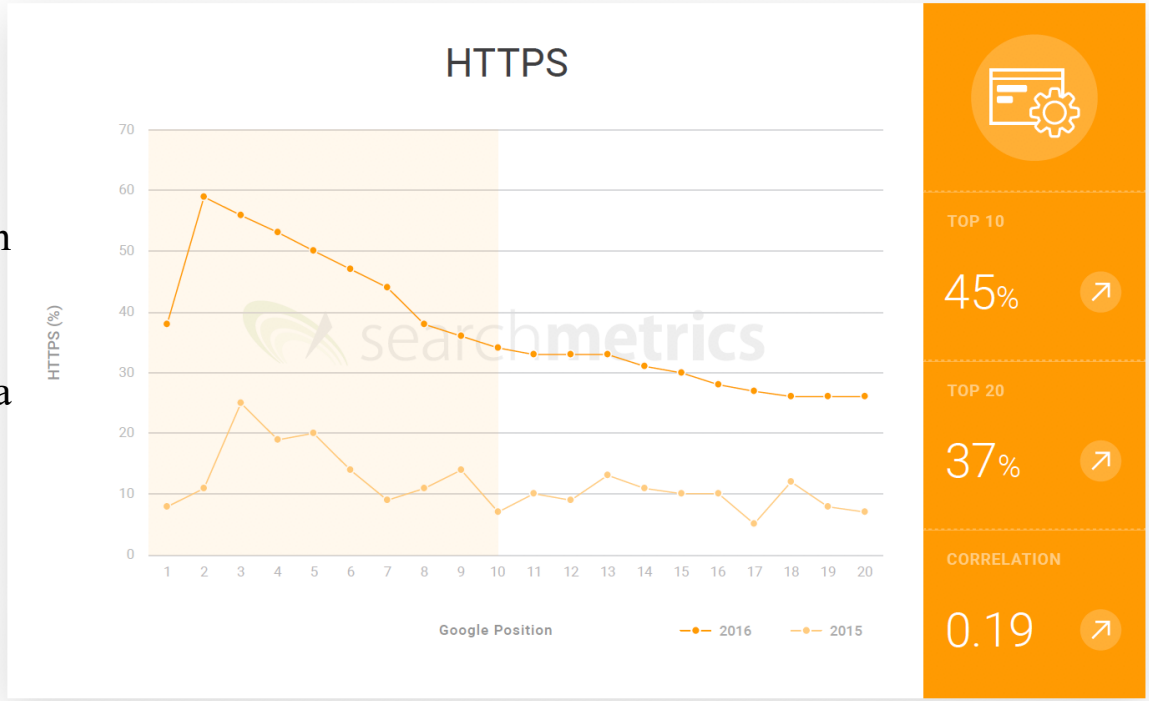
These are single-page sessions where the user leaves the site without interacting with the page.



*SERP: Search Engine Result Page

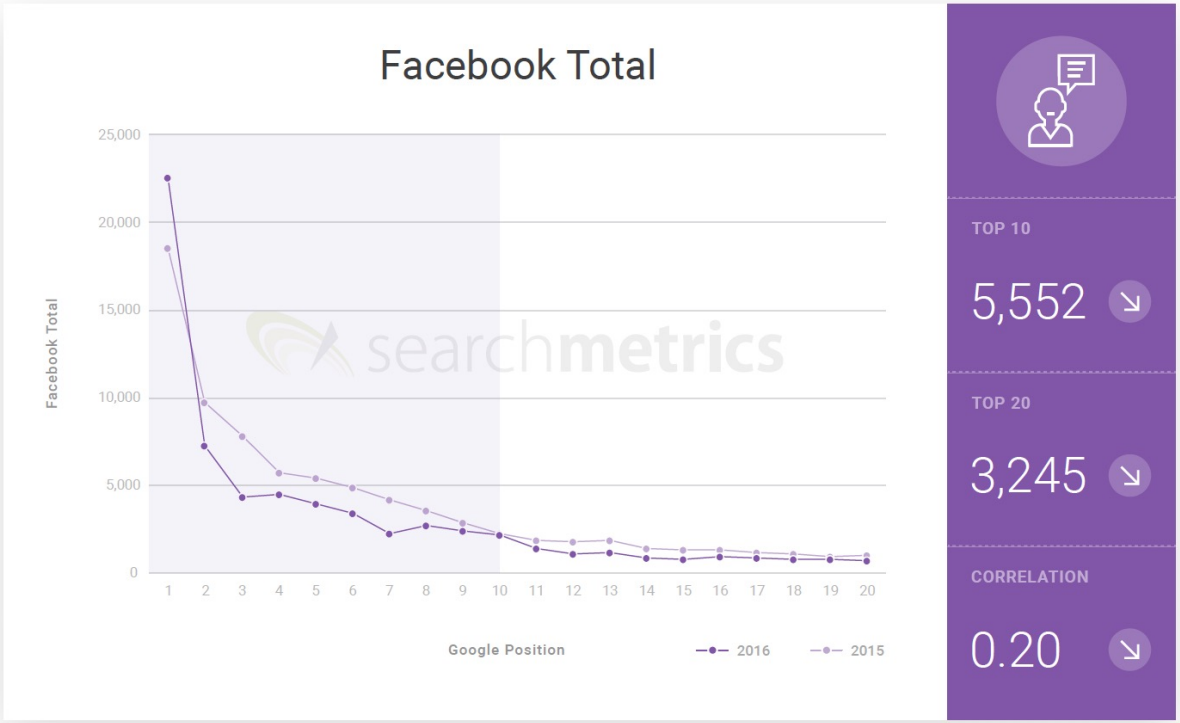
Technical Factors

- 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;

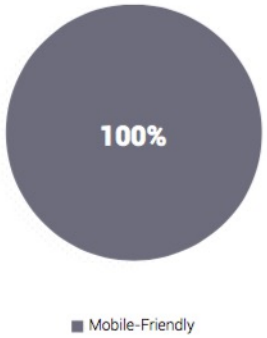
Separate mobile websites are diminishing in popularity, but e.g. try Sephora.com on desktop and mobile. Last time I looked they were still using m.sephora.com

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

Mobile-friendly websites

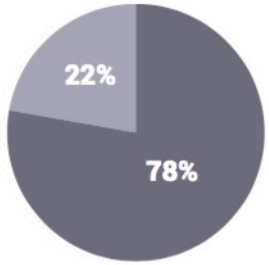
The following graphics show the frequency of websites with mobile-friendly solutions amongst the top 100 domains by SEO visibility.

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://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**

Another Way to Reverse Engineer Google's Query Processing Algorithm

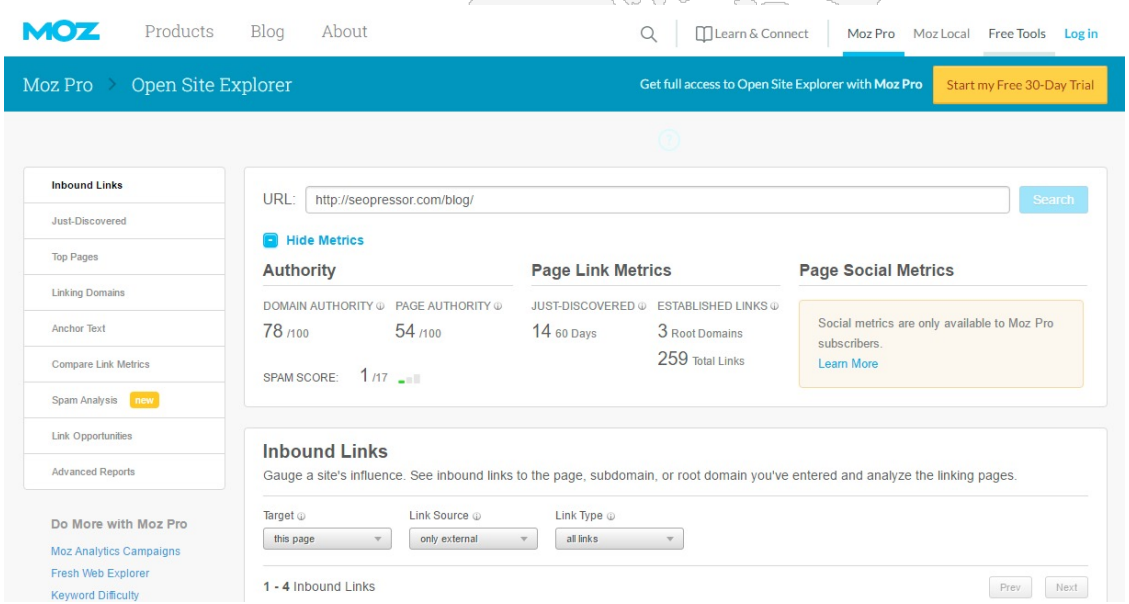
- **Moz.com Monitors Google's Ranking Algorithm watching how search results are affected**
- **Google has changed its ranking algorithm many times over the years**
- **This algorithm is especially important to advertisers**
- **Moz.com is an SEO company that sells its software as a service (SaaS); capabilities 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**

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



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

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

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.

The Google Architecture

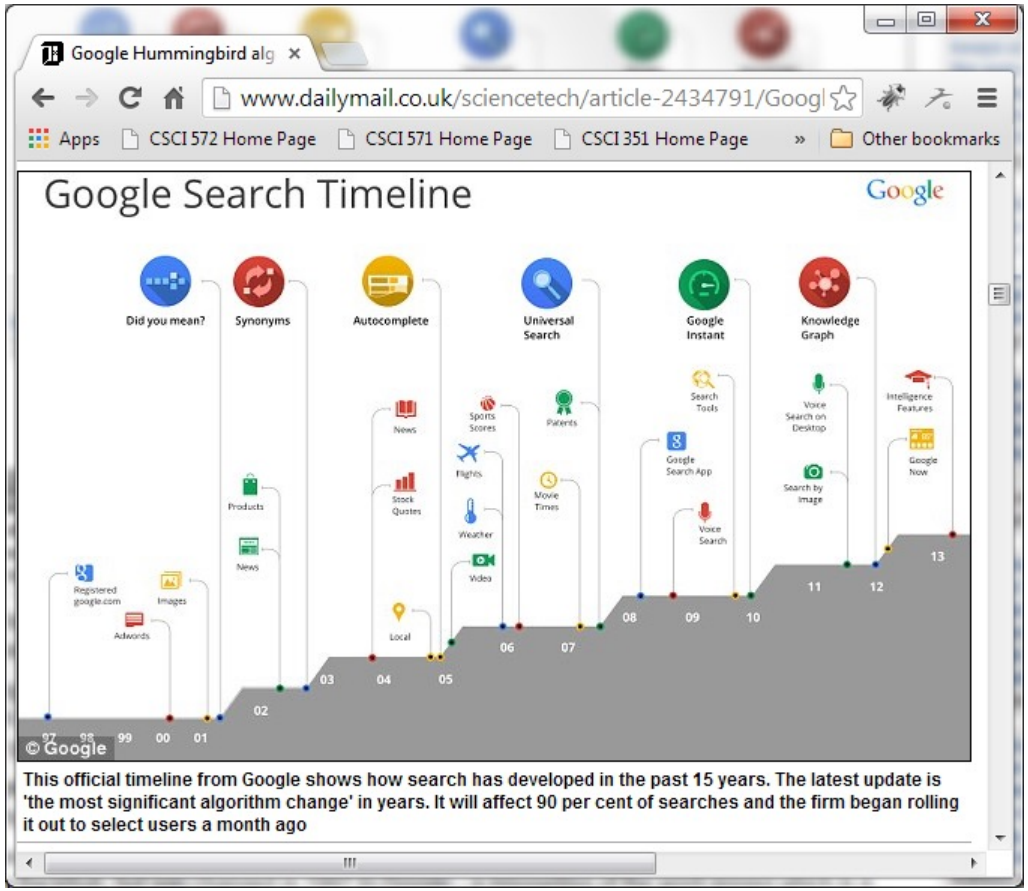
See Google's Website on how search works at

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

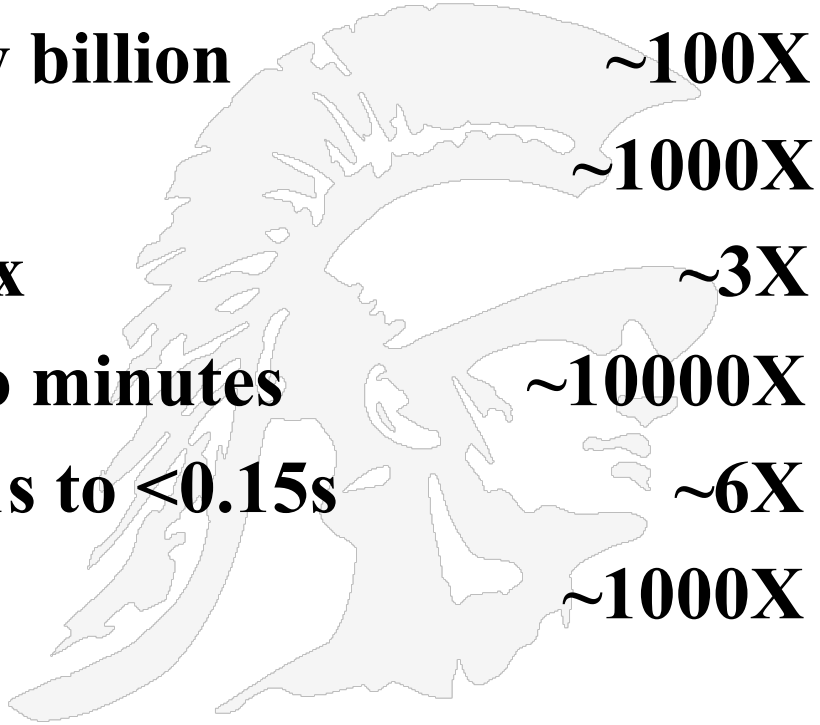


Information Retrieval Challenges for Google from 1999 to 2014

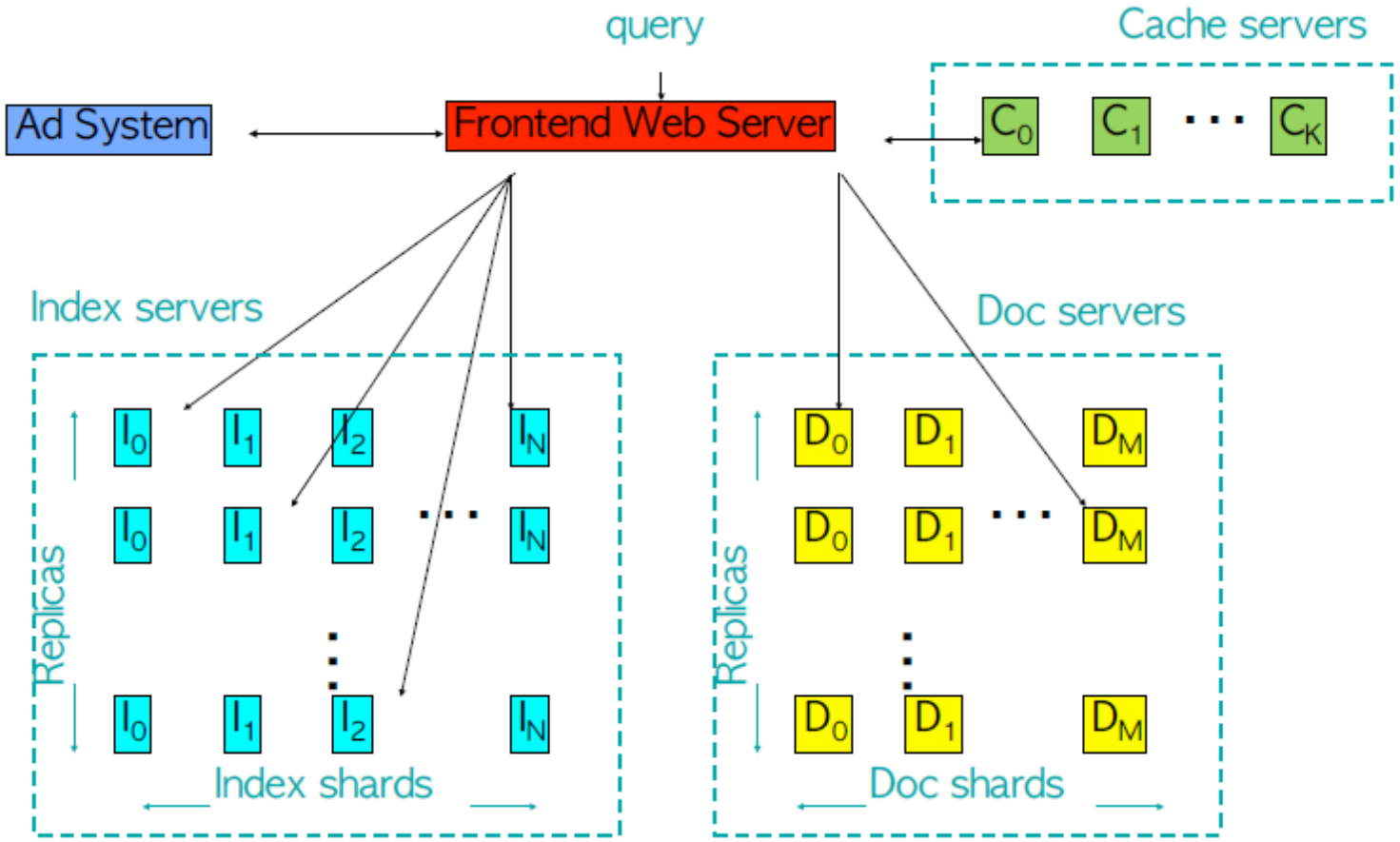
- According to Jeff Dean, Google fellow

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



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

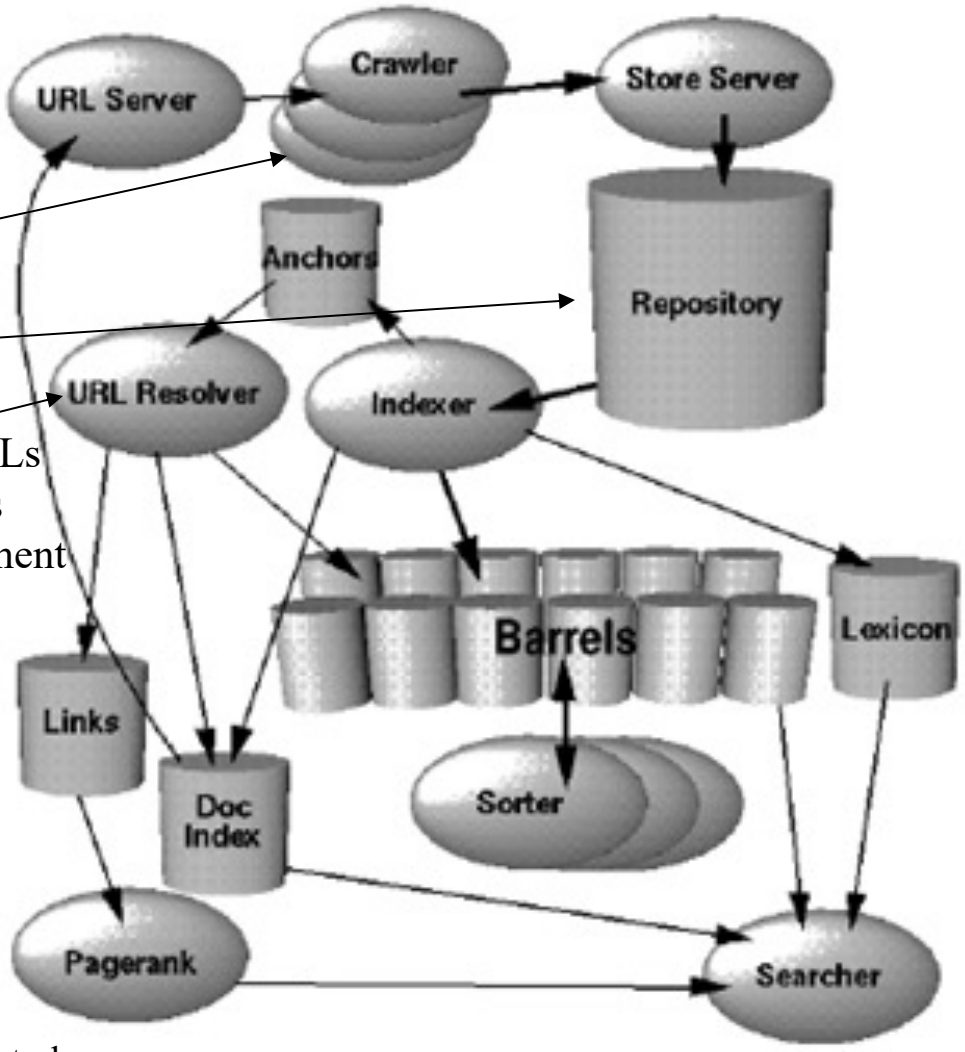


Figure 1. High Level Google Architecture

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

Google'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.

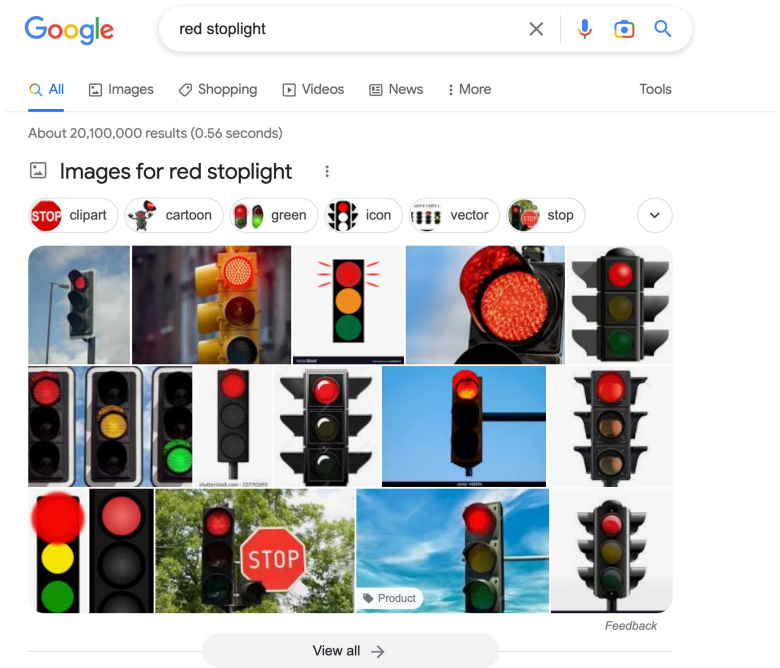
1. Understanding the search query (Search Query Processing)
2. Relevance determination at document level (scoring)
3. Compilation of search results and SERP* features

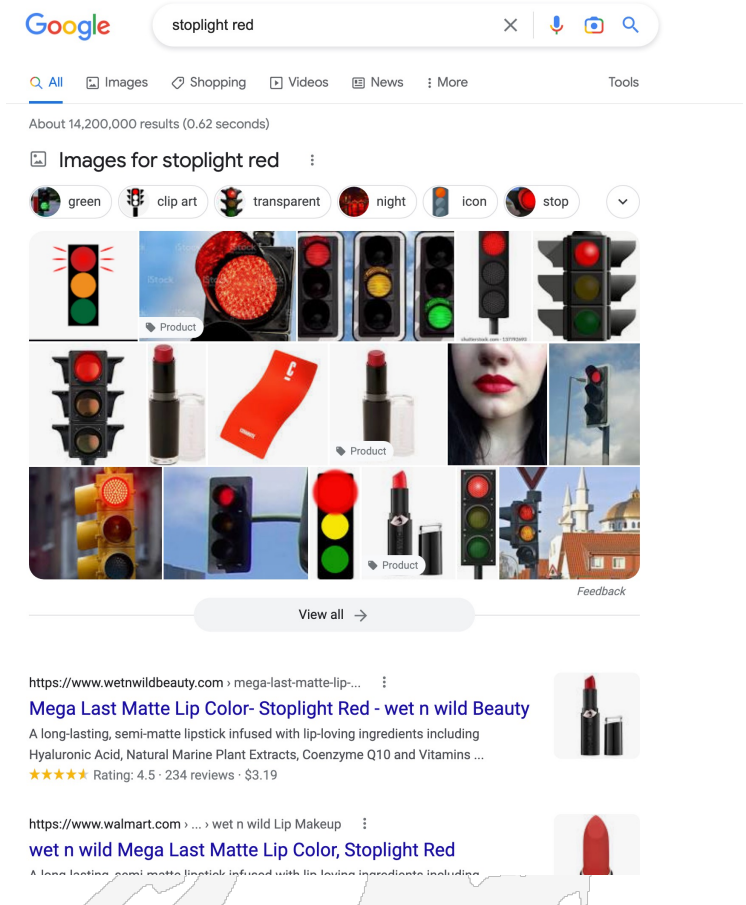
- ***Important steps***

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

Term-Based versus Entity-Based Meaning

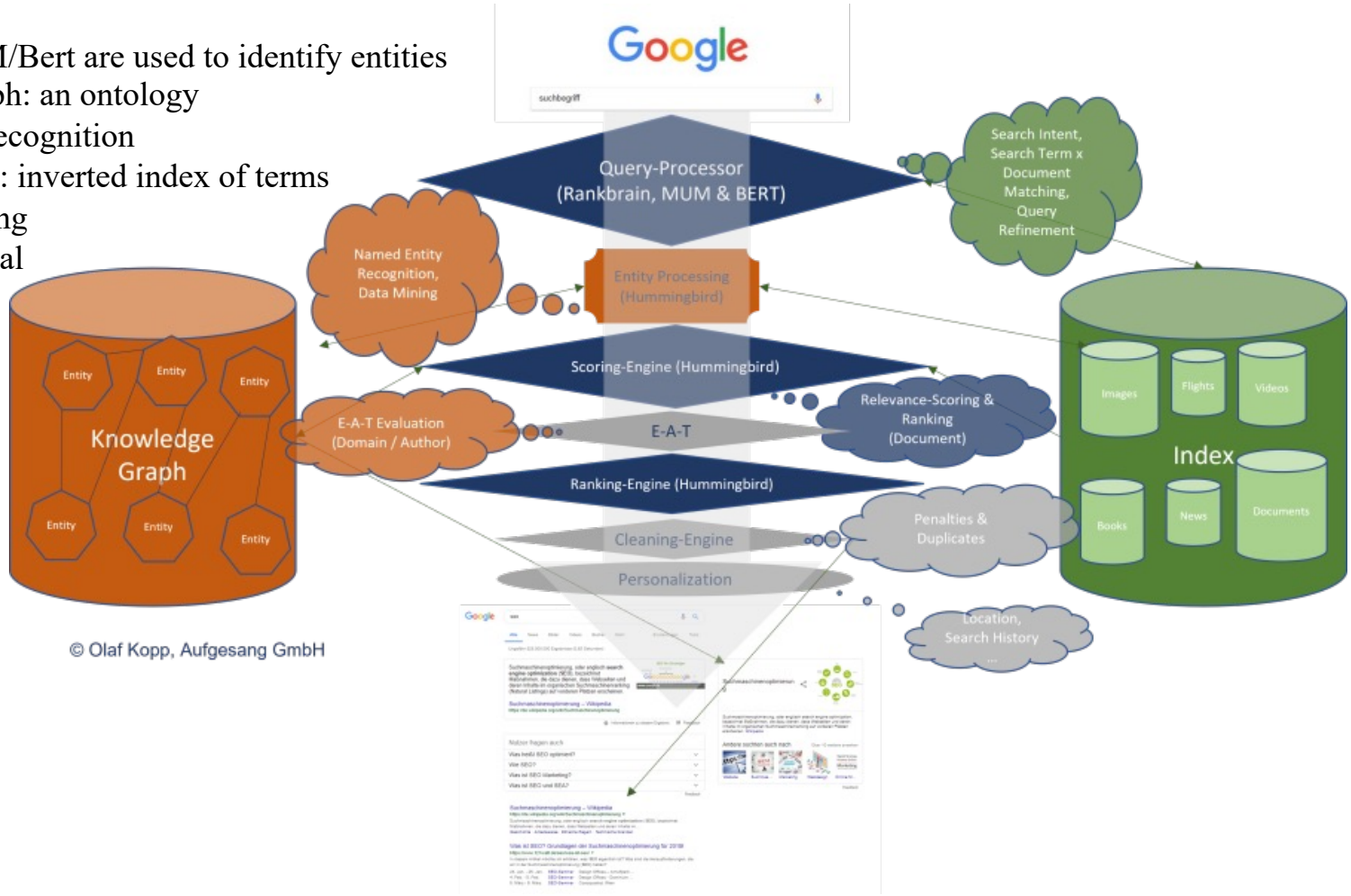




An entity based search recognizes the different context based on the different arrangement; "Stoplight red" is an entity, not to be confused as separate terms "red" and "stoplight"

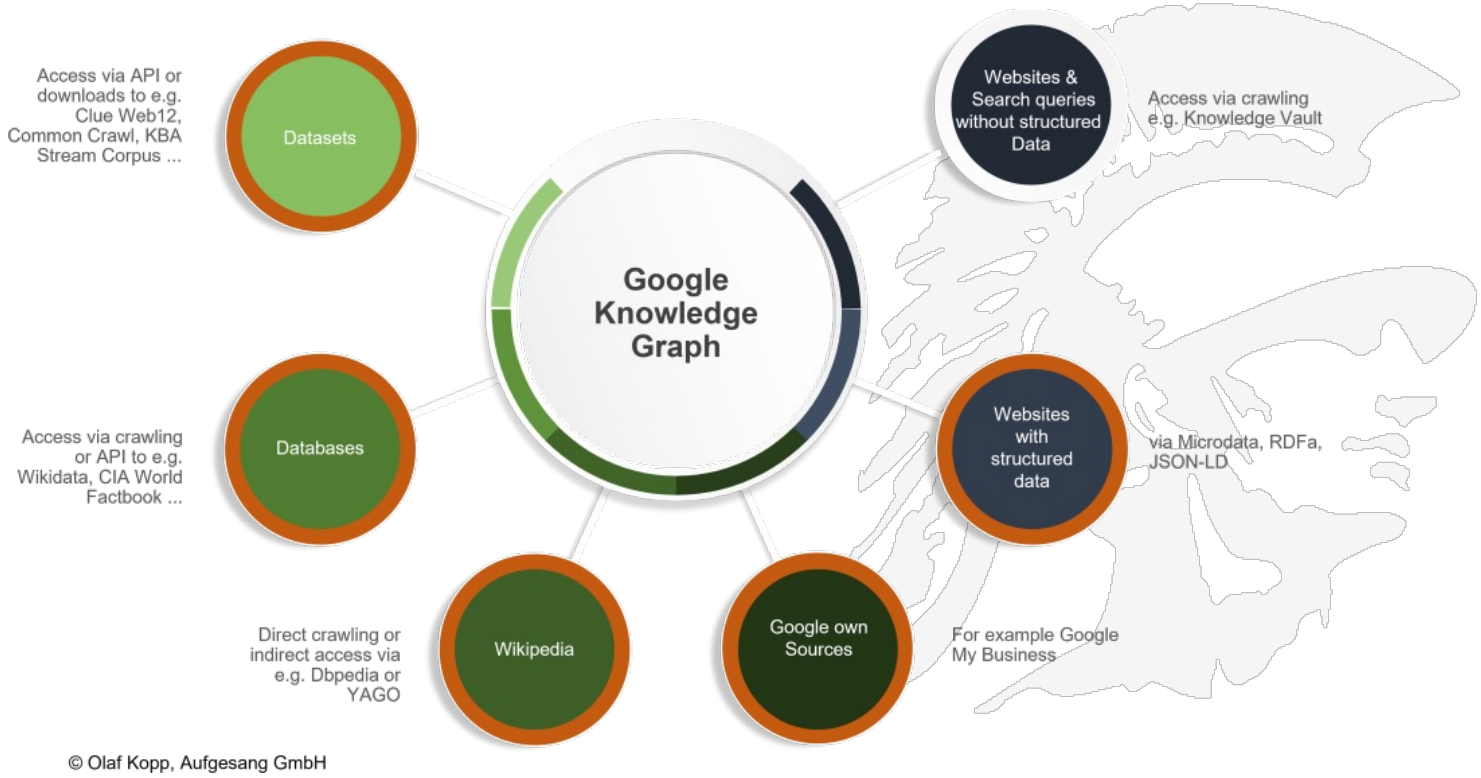
Google's Query Processing Elements

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



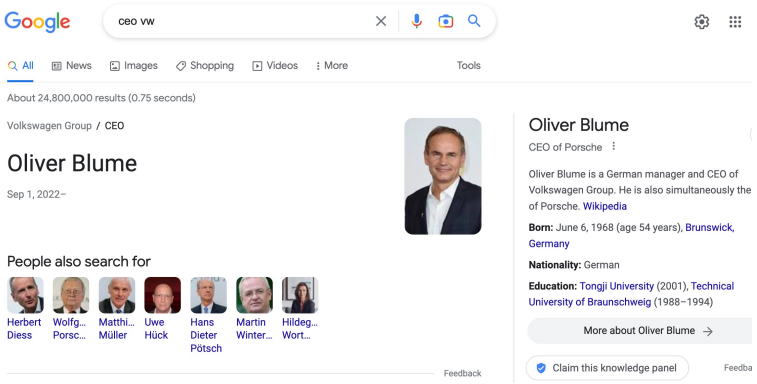
Using the KnowledgeGraph to Identify Entities

- 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.

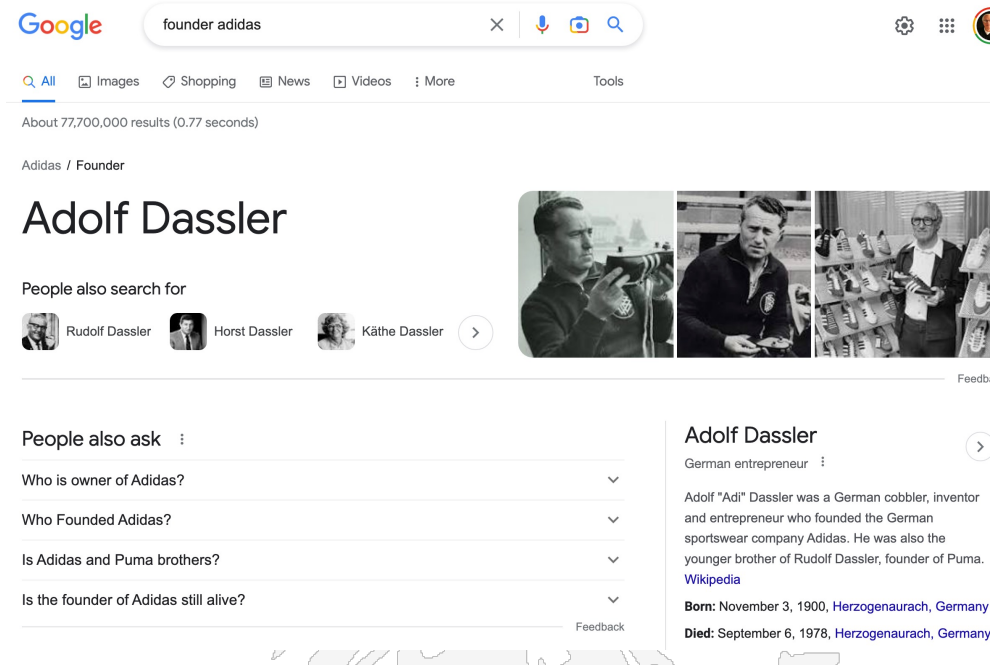


USC Viterbi School of Engineering

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"

Google Patent Describing Semantic Clustering

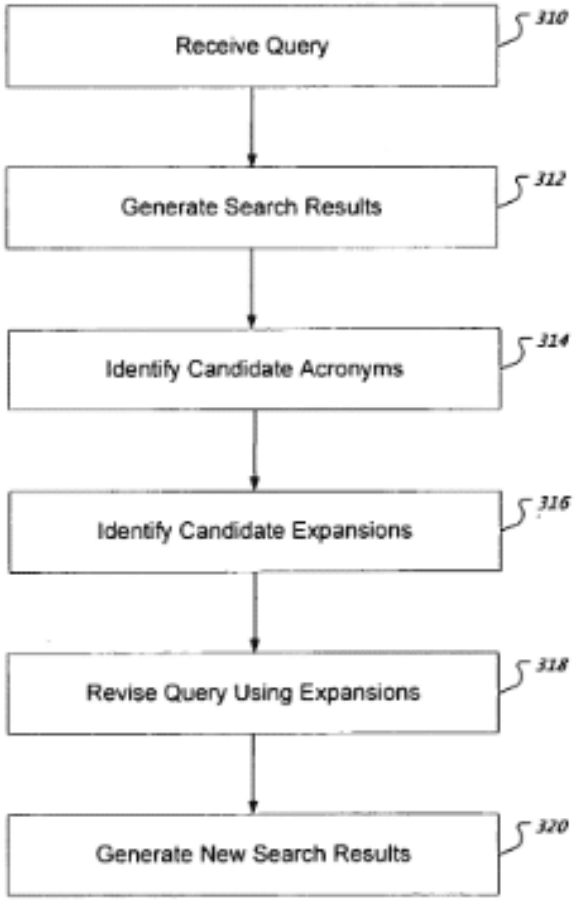
- Terms are assigned to entities and to a thematic context
  - E.g. the term "mercedes" to ID3279 and its context is "Auto"
- Patent title: *Improving topic clustering on search queries with word co-occurrence and bipartite graph co-clustering* is based on this paper
  - https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45569.pdf
  - The notion of "lift" scores is introduced using this formula

$$lift(\mathrm{w}_i, \mathbf{a}) = \frac{P(\mathrm{w}_i \mid \mathbf{a})}{P(\mathrm{w}_i)},$$

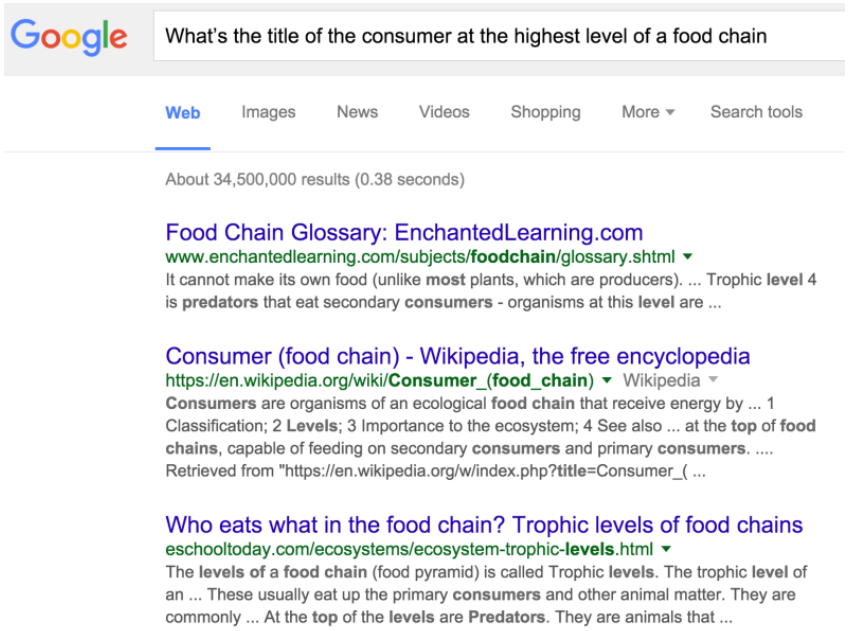
- w_i is in the formula for all terms that are closely related to the root of the word, such as misspellings, plural, singular or synonyms.
- ***"a"*** can be any user interaction such as searching for a specific search term or visiting a specific page.
- For example, if the lift score is 5, the probability that "Wi" will be searched is 5 times higher than that of "Wi" being searched in general.
-

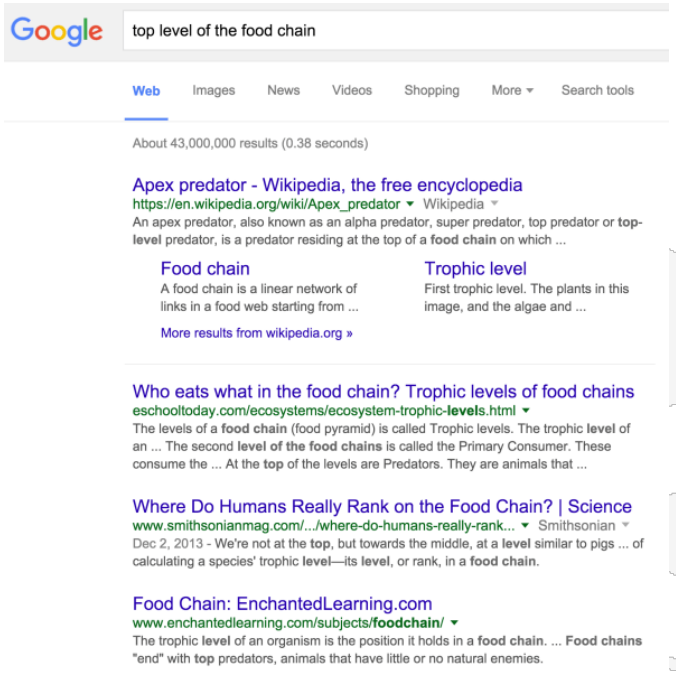
RankBrain
An Entity-Based Processor

- 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

RankBrain in Its Simplest Form

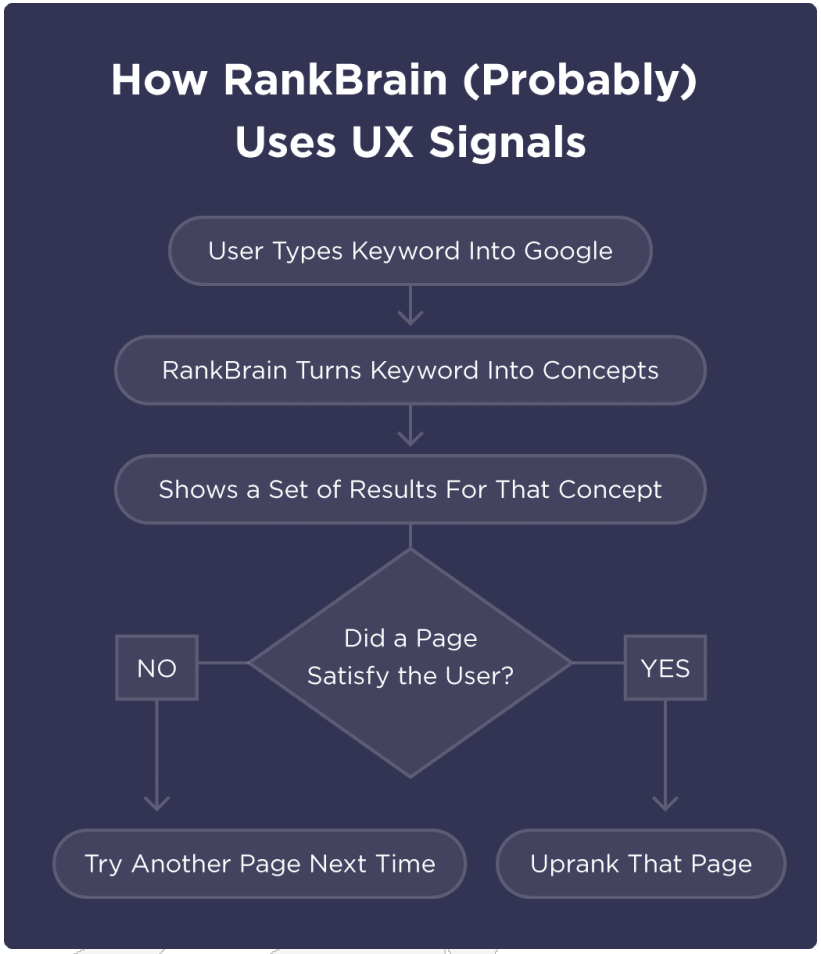
1. Google receives a query for something it's never seen before (like a new movie title or a phrase connecting two topics, like "which country has the best cars")
2. Google assigns the entity a unique identifier, like 9202a8c04000641f8000000000006567.
3. Google determines the entity's relatedness to 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.

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

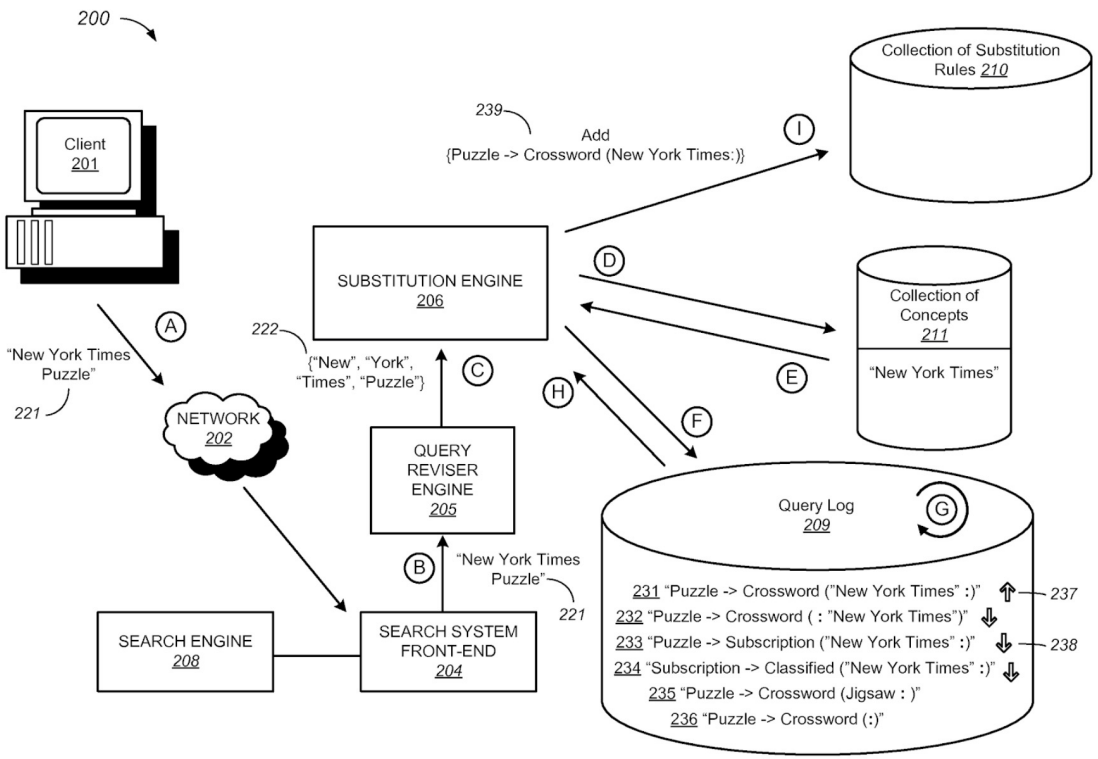
- These are known as user experience signals (UX signals).



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.**



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

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

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