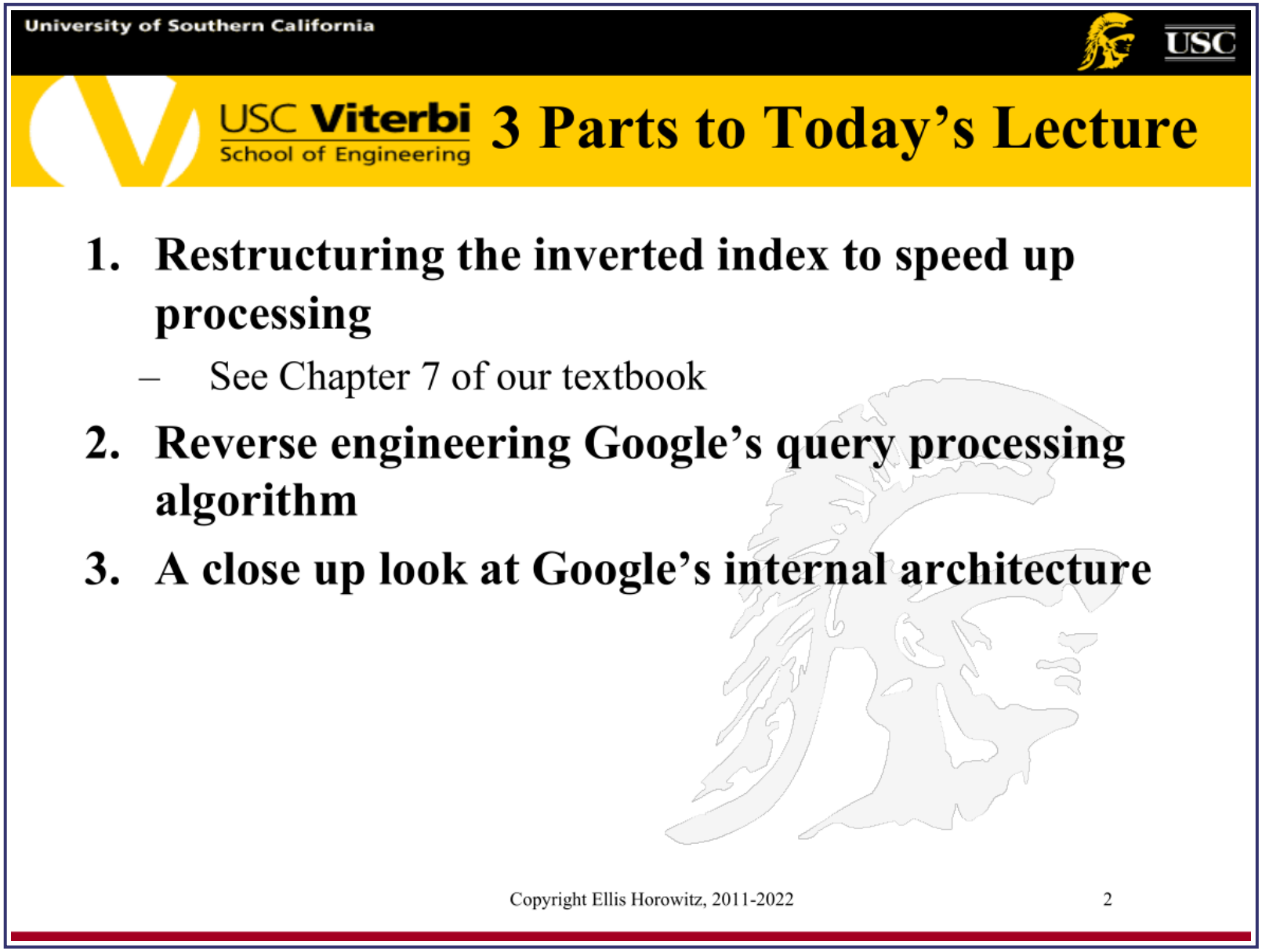
Query processing

how does the search engine respond?

••


University of Southern California
USC
USC Viterbi
School of Engineering
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
2

●●

University of Southern California

USC Viterbi School of Engineering

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**

••

University of Southern California

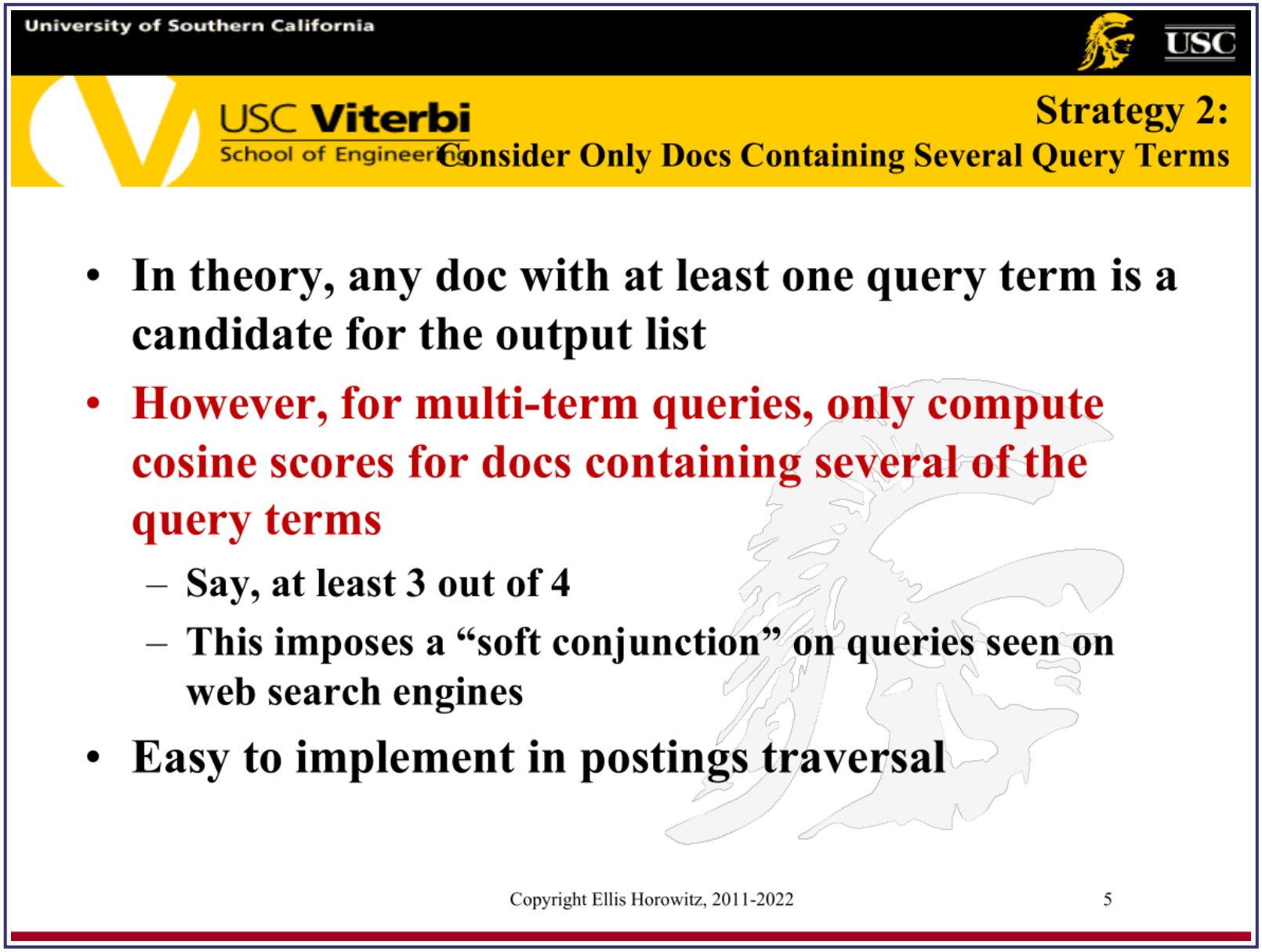
USC Viterbi School of Engineering

USC

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

••


University of Southern California
USC
USC Viterbi
School of Engineering
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
5

••

University of Southern California

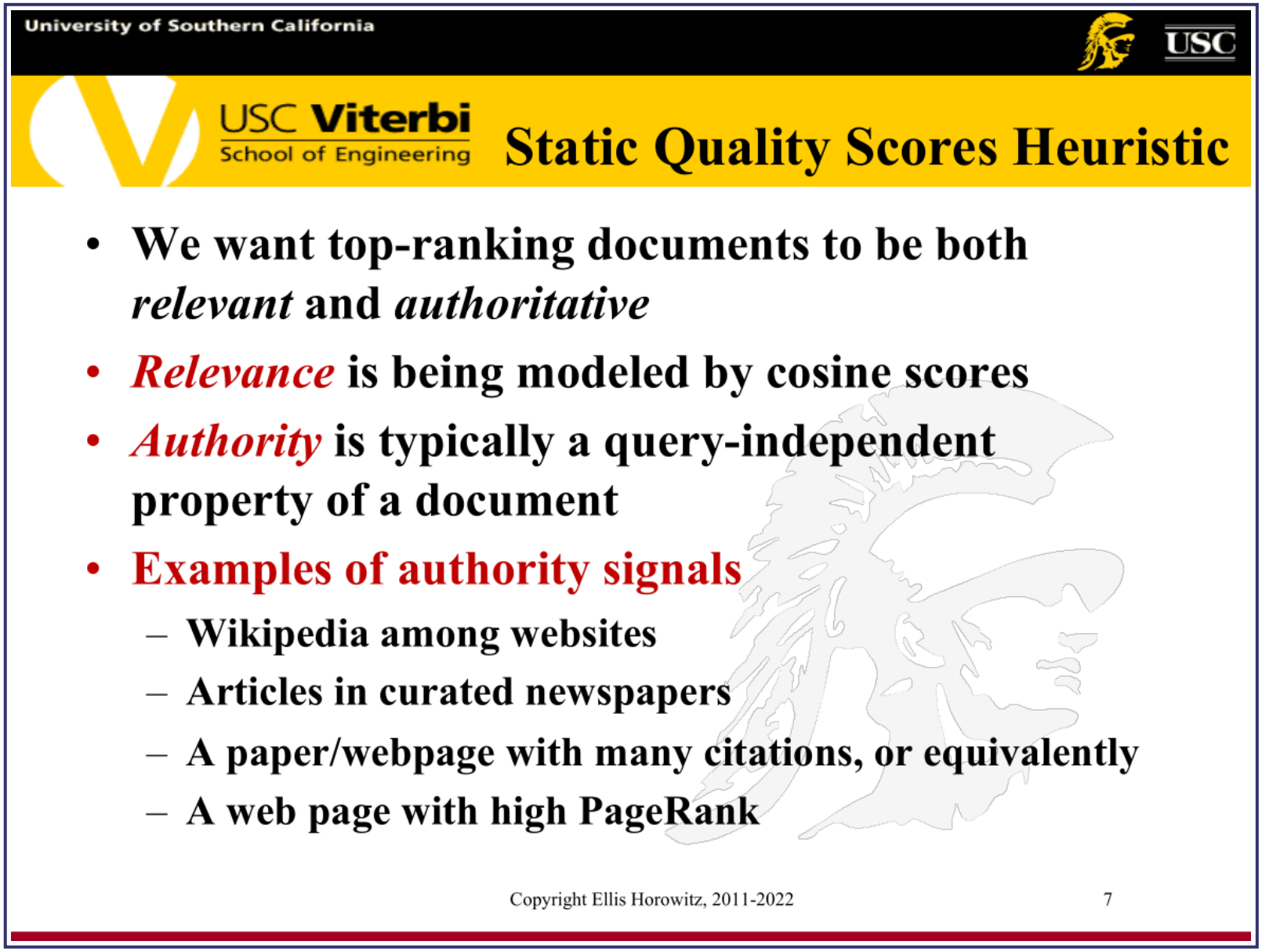
USC Viterbi School of Engineering

USC

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

••


University of Southern California
USC
USC Viterbi
School of Engineering
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
7

●●

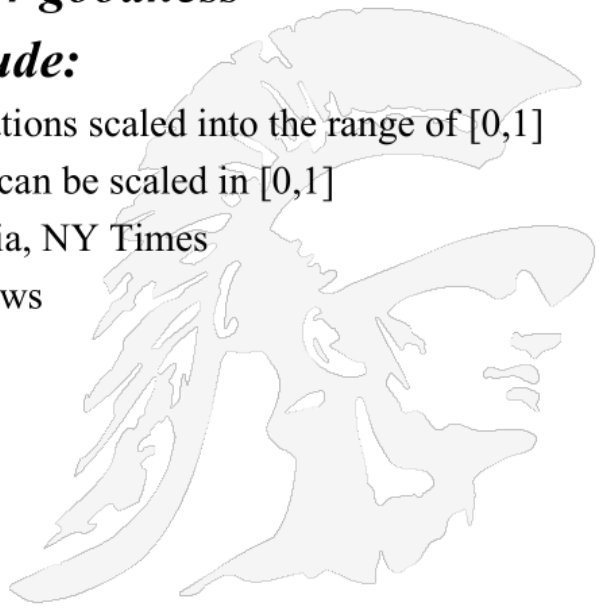
University of Southern California

USC Viterbi School of Engineering

USC

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



••

University of Southern California

USC Viterbi School of Engineering

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

••

University of Southern California

USC Viterbi School of Engineering

USC

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

••

University of Southern California

USC Viterbi School of Engineering

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

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

••

University of Southern California

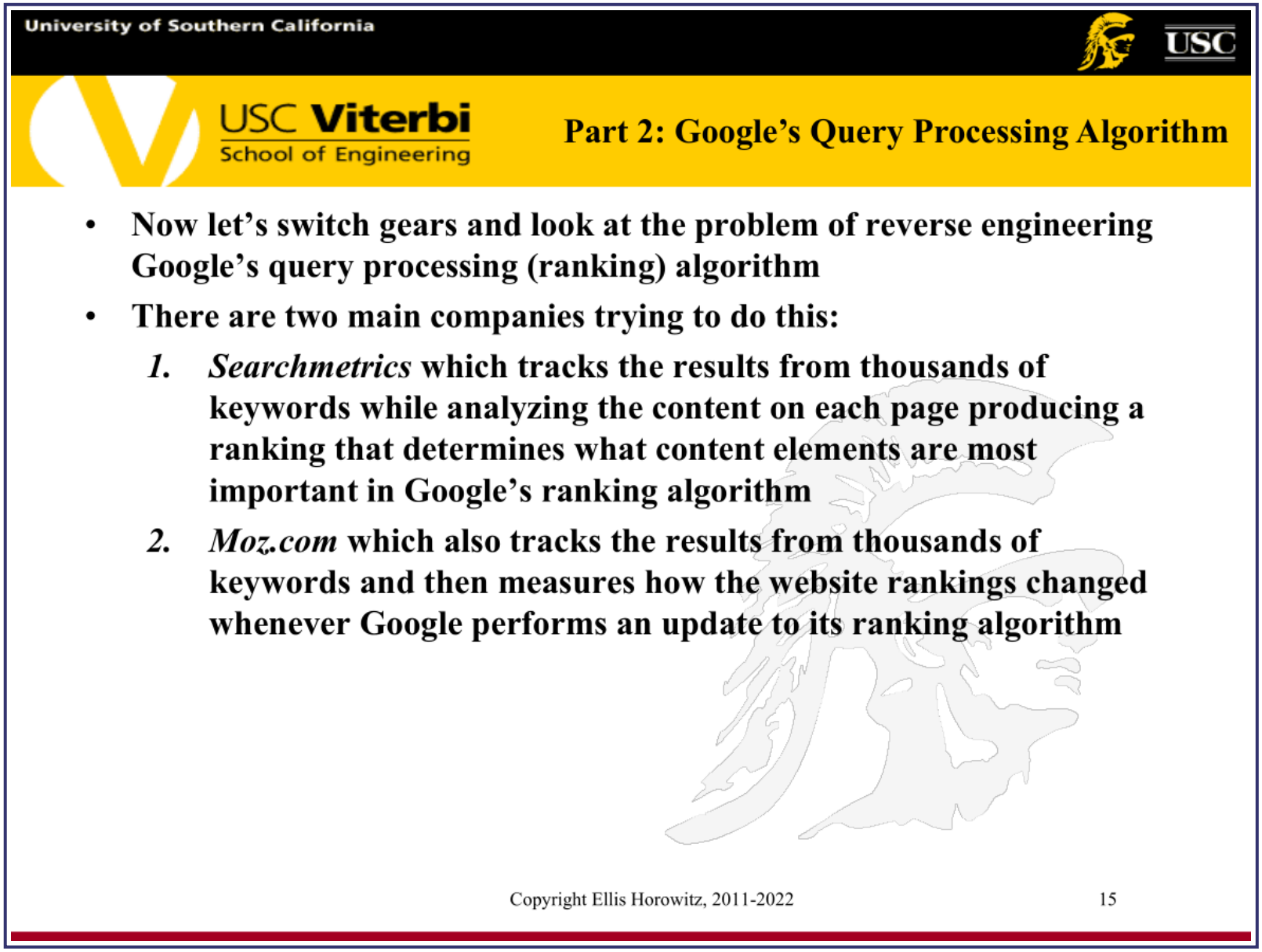
USC Viterbi School of Engineering

USC

Strategy 6: High and Low Lists Heuristic

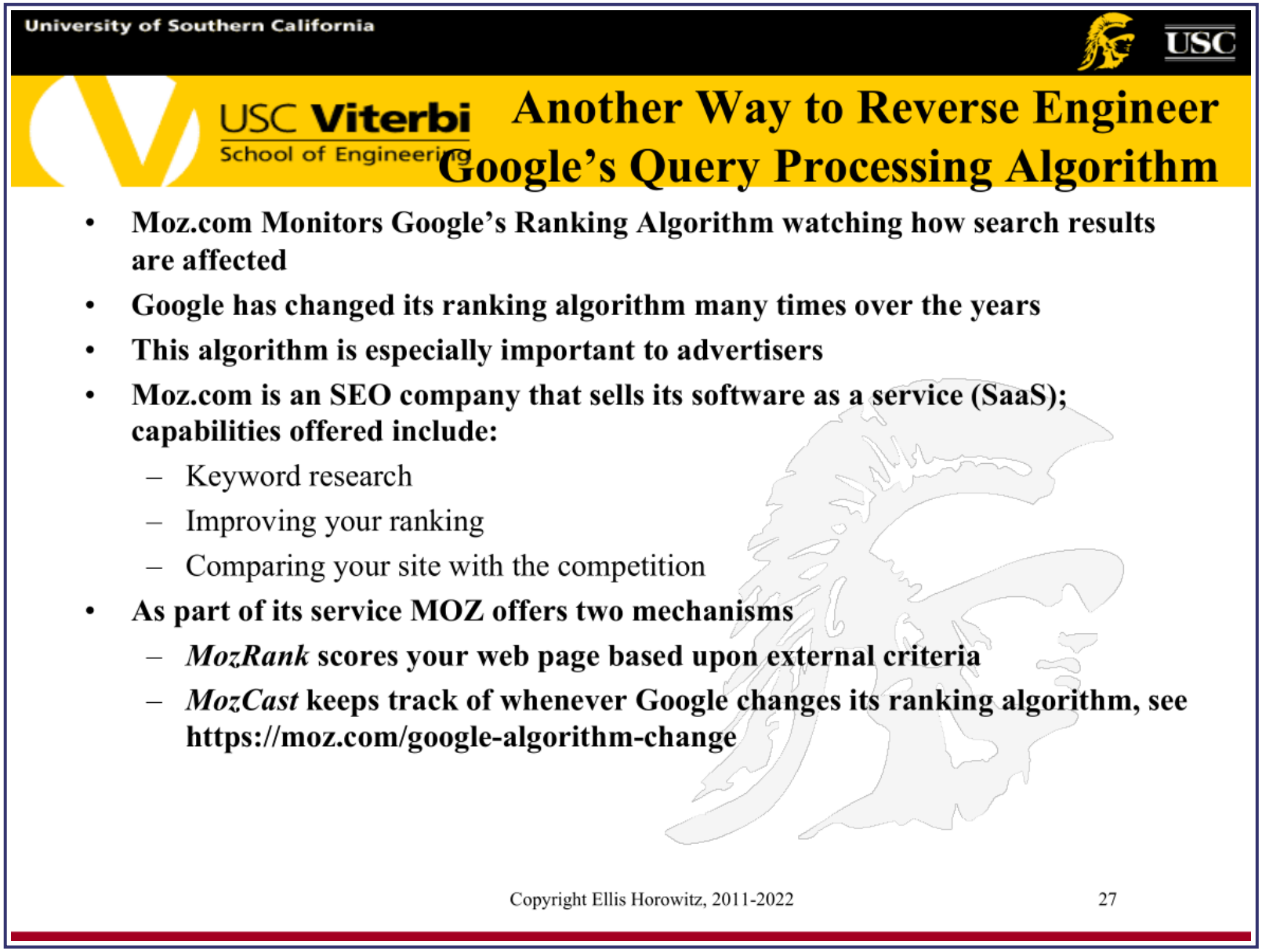
- **For each term, maintain two postings lists called *high* and *low***
  - Think of *high* as the champion list
- **When traversing postings on a query, only traverse *high* lists first**
  - If we get more than K docs, select the top K and stop
  - Else proceed to get docs from the *low* lists
- **Can be used even for simple cosine scores, without global quality *g(d)***
- **This assumes we have a means for segmenting index into two tiers**

●●


University of Southern California
USC
USC Viterbi
School of Engineering
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
15

This is Searchmetrics' doc that lists the various metrics ('ranking factors') they use, to characterize Google's page ranking algorithm.

●●


University of Southern California
USC
USC Viterbi
School of Engineering
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
27

This page at moz.com lists every change in Google's SERP (results page) influencing algorithms, going back to the year 2000.

●●

University of Southern California

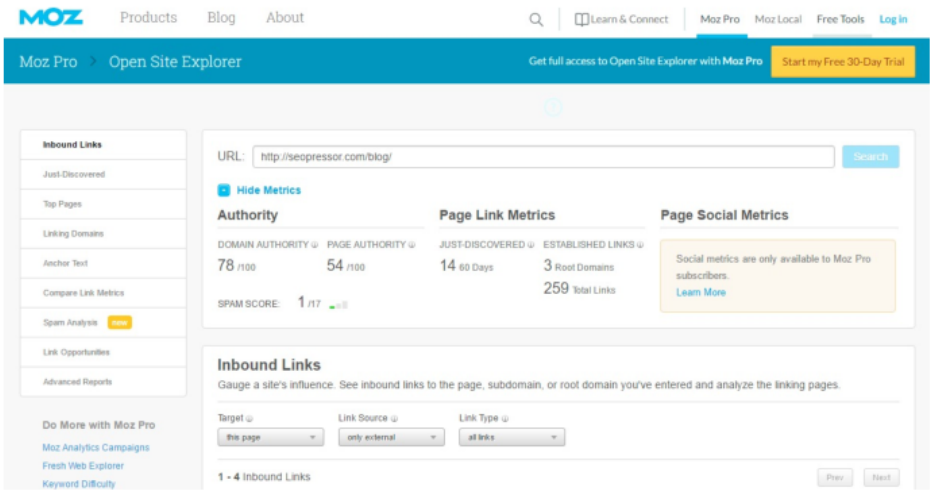
USC Viterbi School of Engineering

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



Here is Moz's link explorer page.

●●

University of Southern California

USC Viterbi School of Engineering

USC

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.

●●


University of Southern California
USC
USC Viterbi
School of Engineering
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

This is the 'how search works' page.

●●

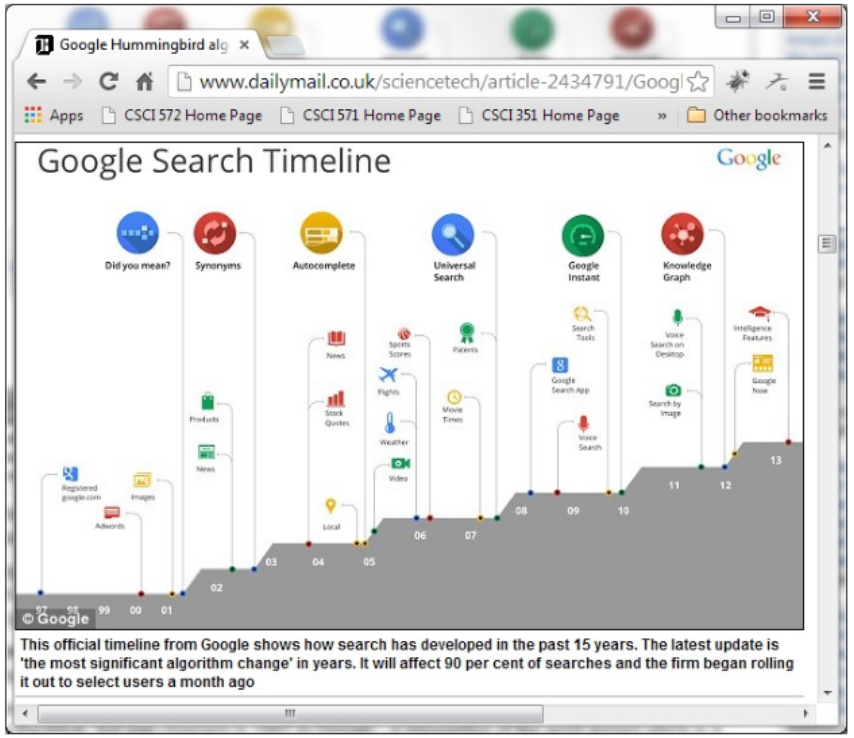
University of Southern California

USC

USC Viterbi School of Engineering

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

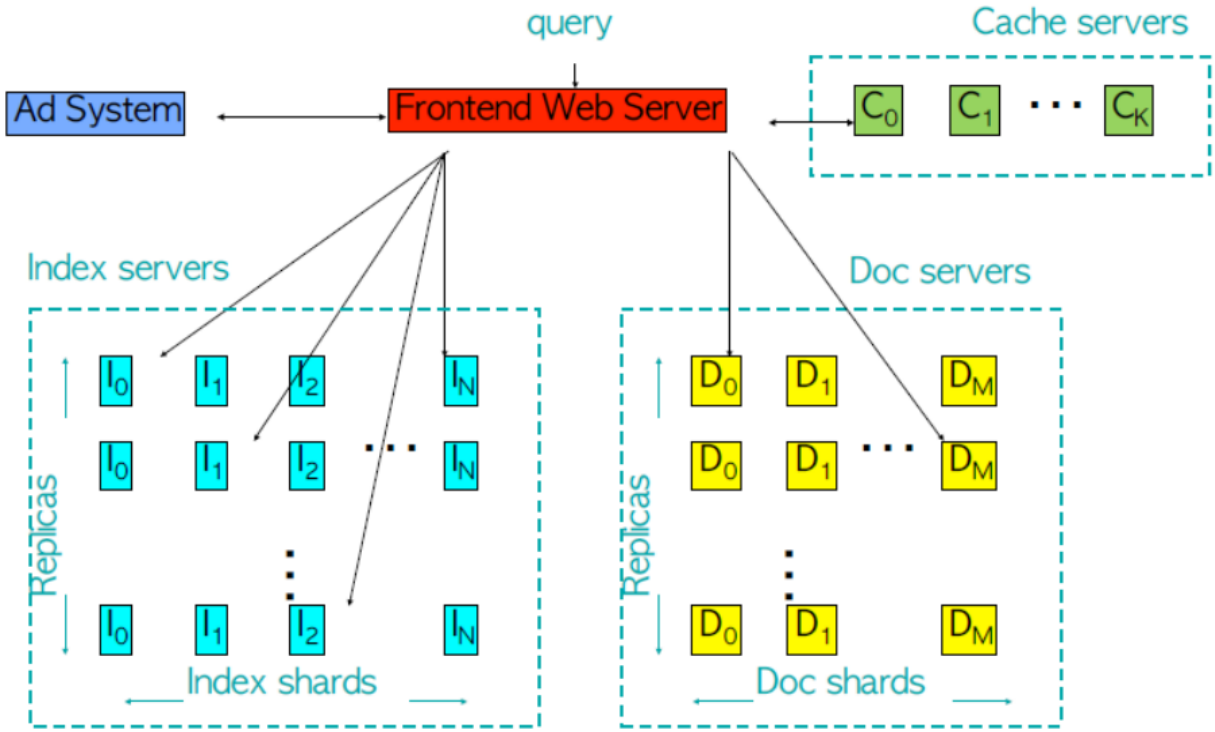


●●

University of Southern California

USC Viterbi School of Engineering

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.

●●

University of Southern California

USC Viterbi School of

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

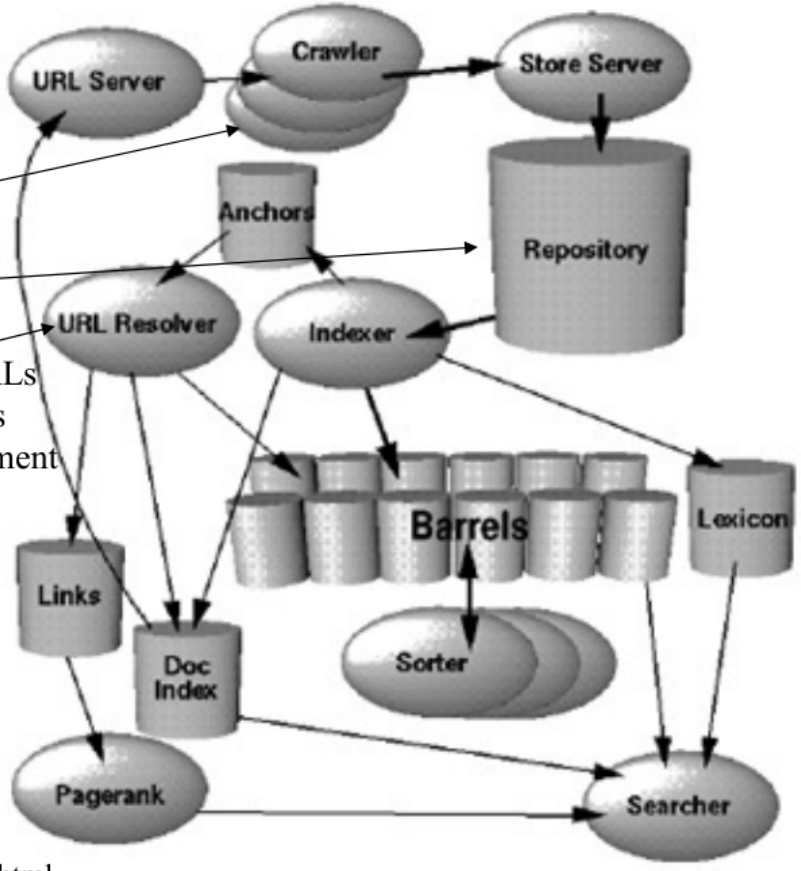
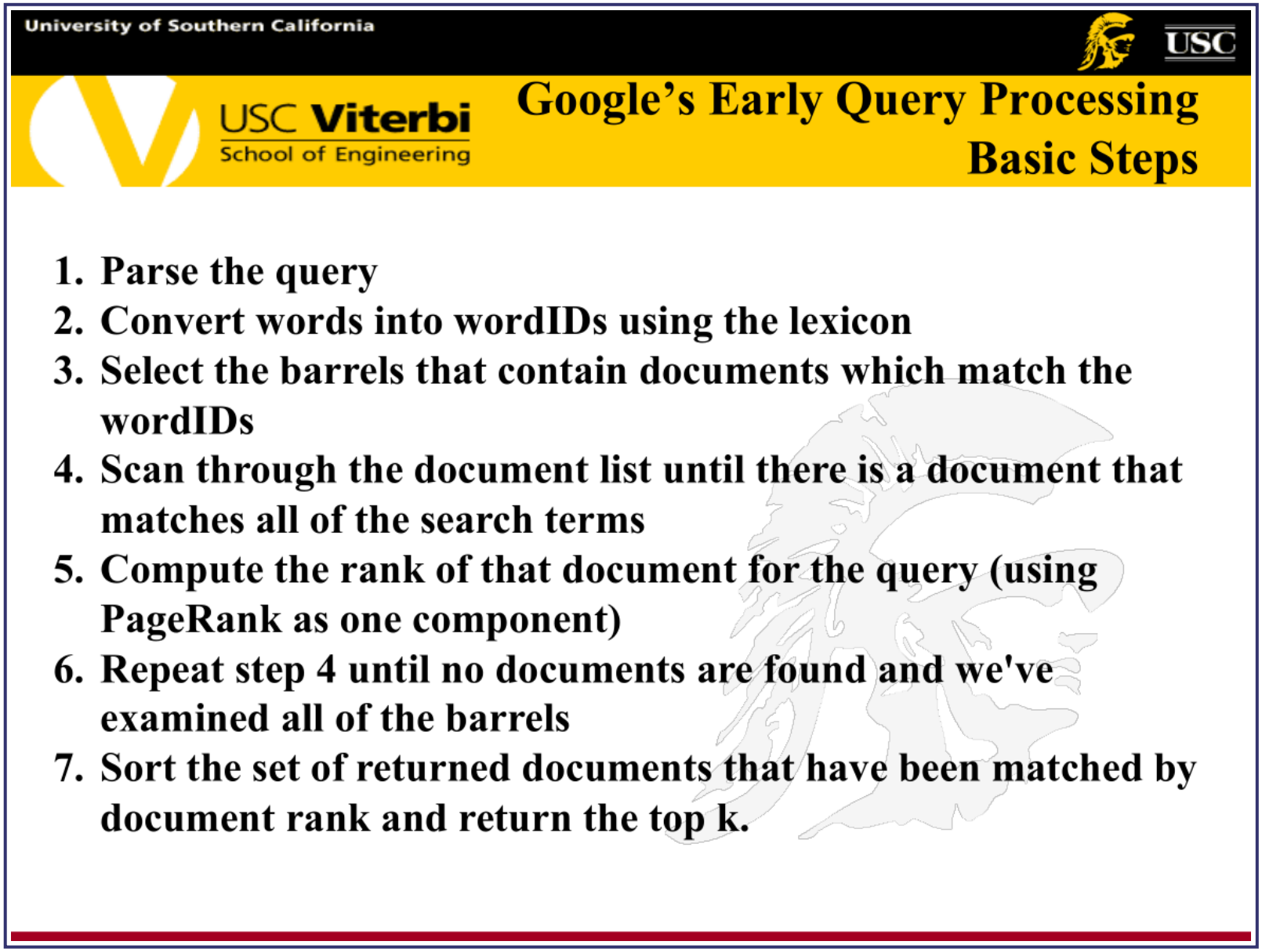


Diagram from
"*The Anatomy of a Large-Scale Hypertextual Web Search Engine*"
http://infolab.stanford.edu/~backrub/google.html

Figure 1. High Level Google Architecture

••


University of Southern California
USC
USC Viterbi
School of Engineering
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.

●●

University of Southern California

USC Viterbi School of Engineering

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

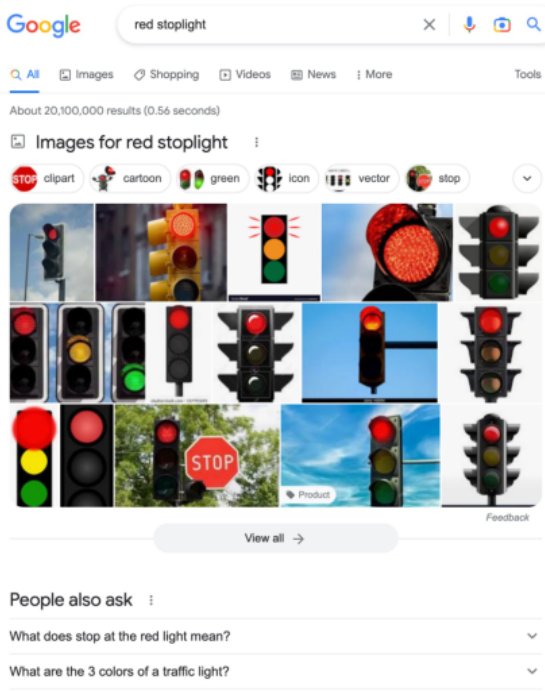
- *Search engine results page

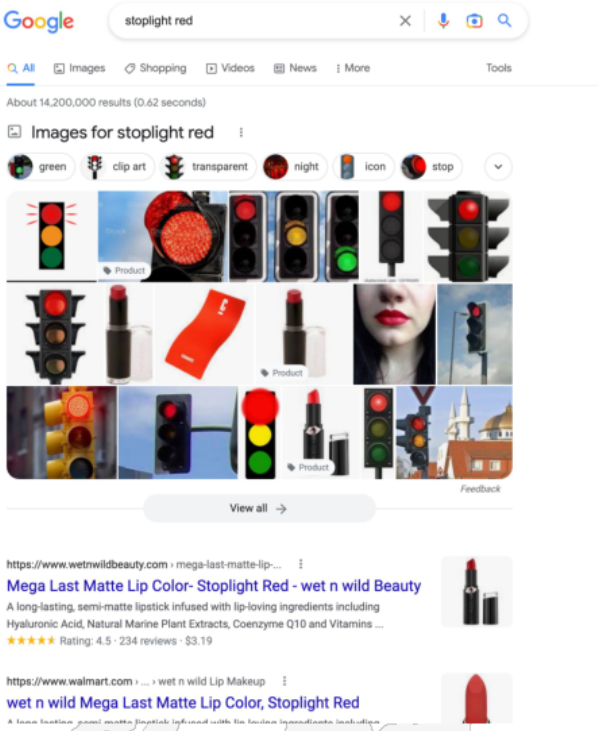
●●

University of Southern California

USC Viterbi School of Engineering

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"

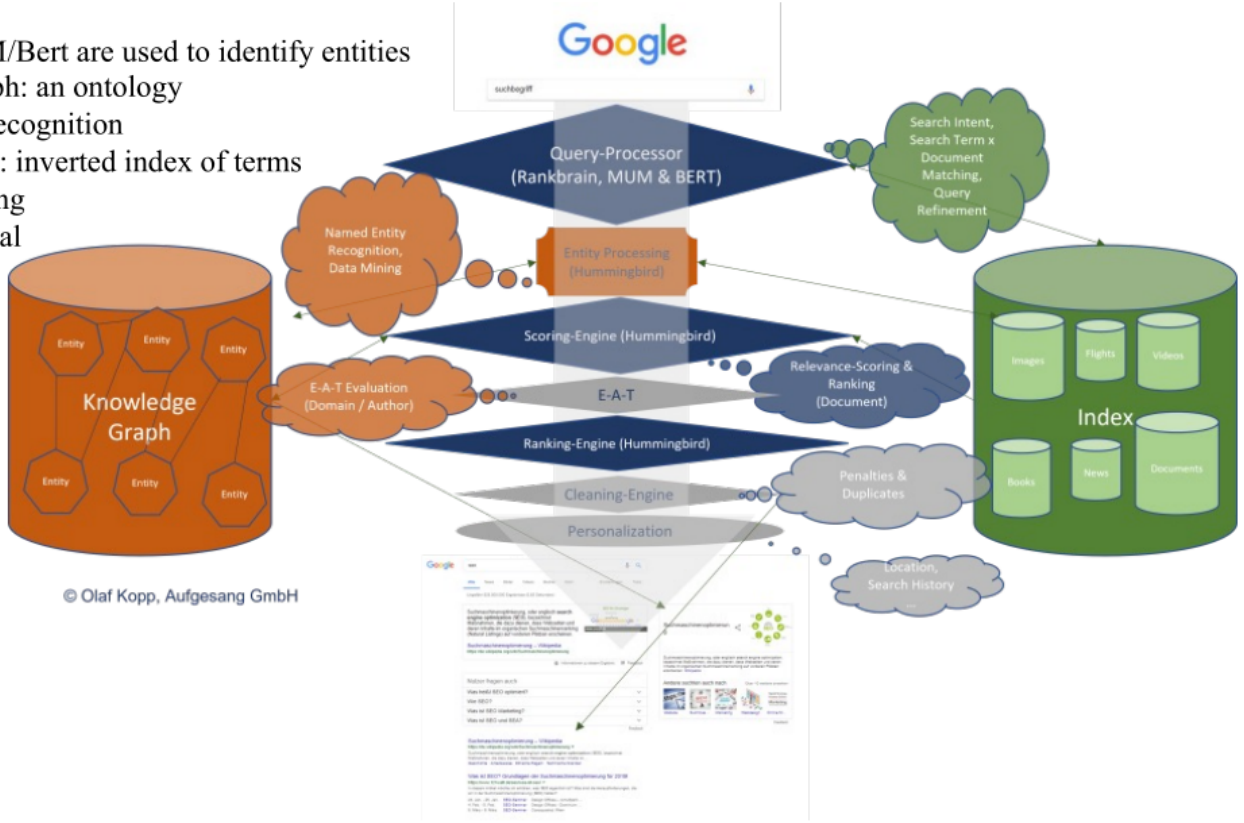
●●

University of Southern California

USC Viterbi School of Engineering

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



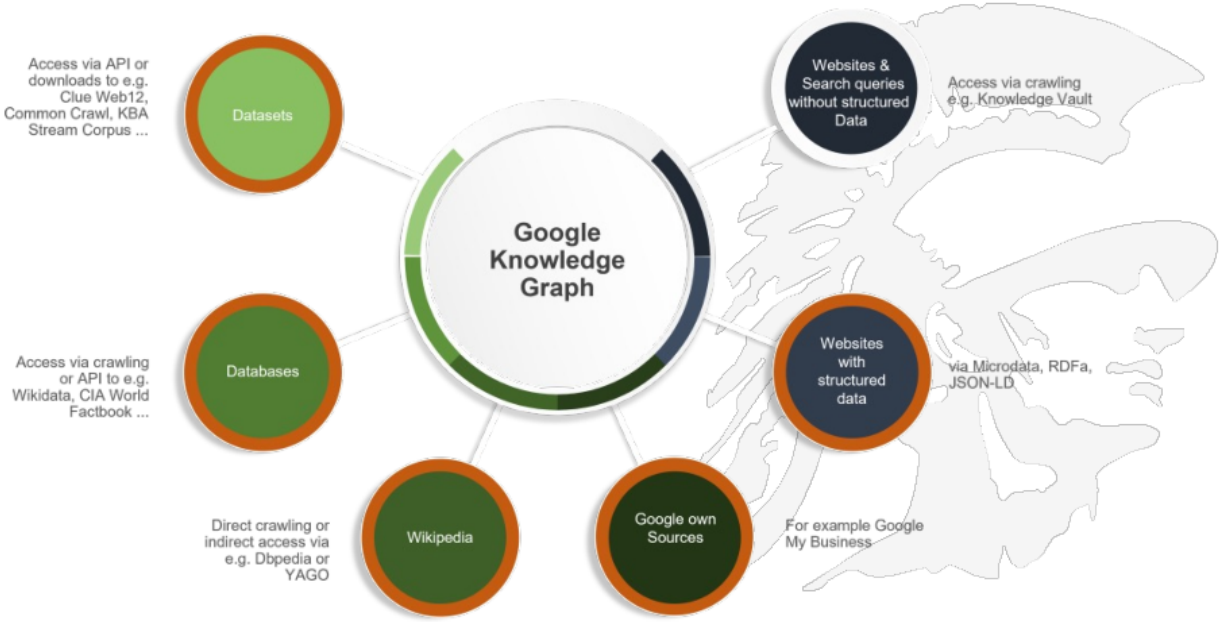
●●

University of Southern California

USC Viterbi School of Engineering

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.



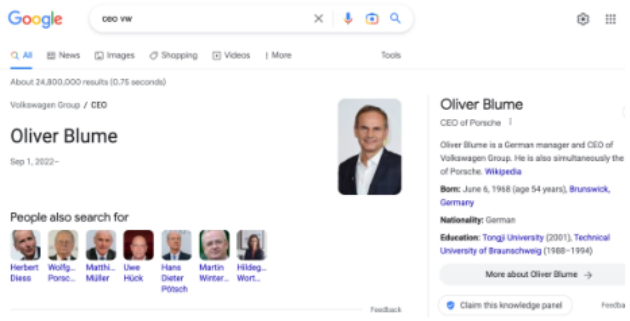
 41

●●

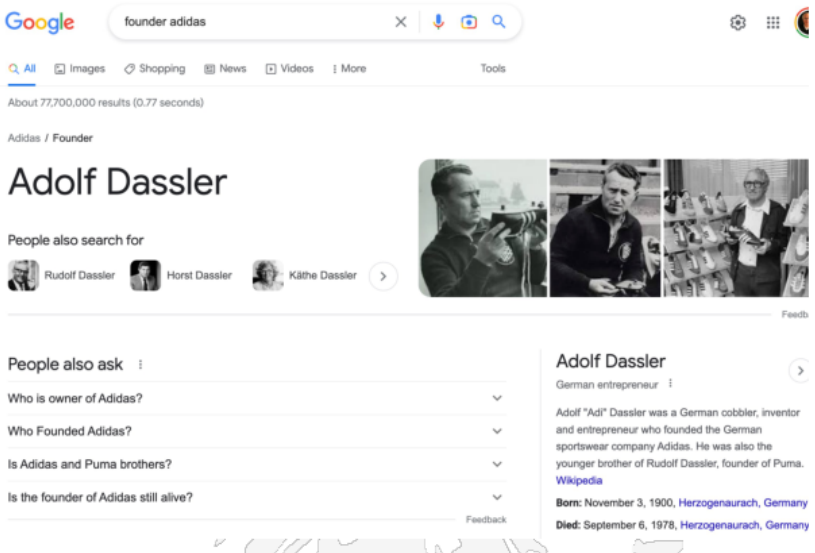
University of Southern California

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"

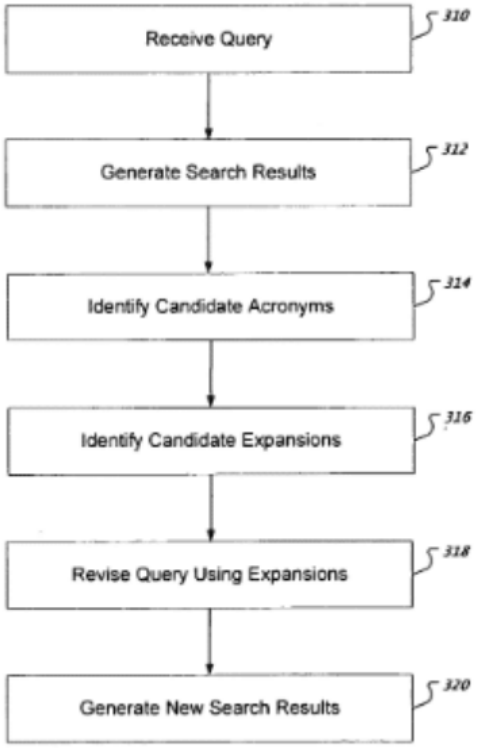
●●

University of Southern California

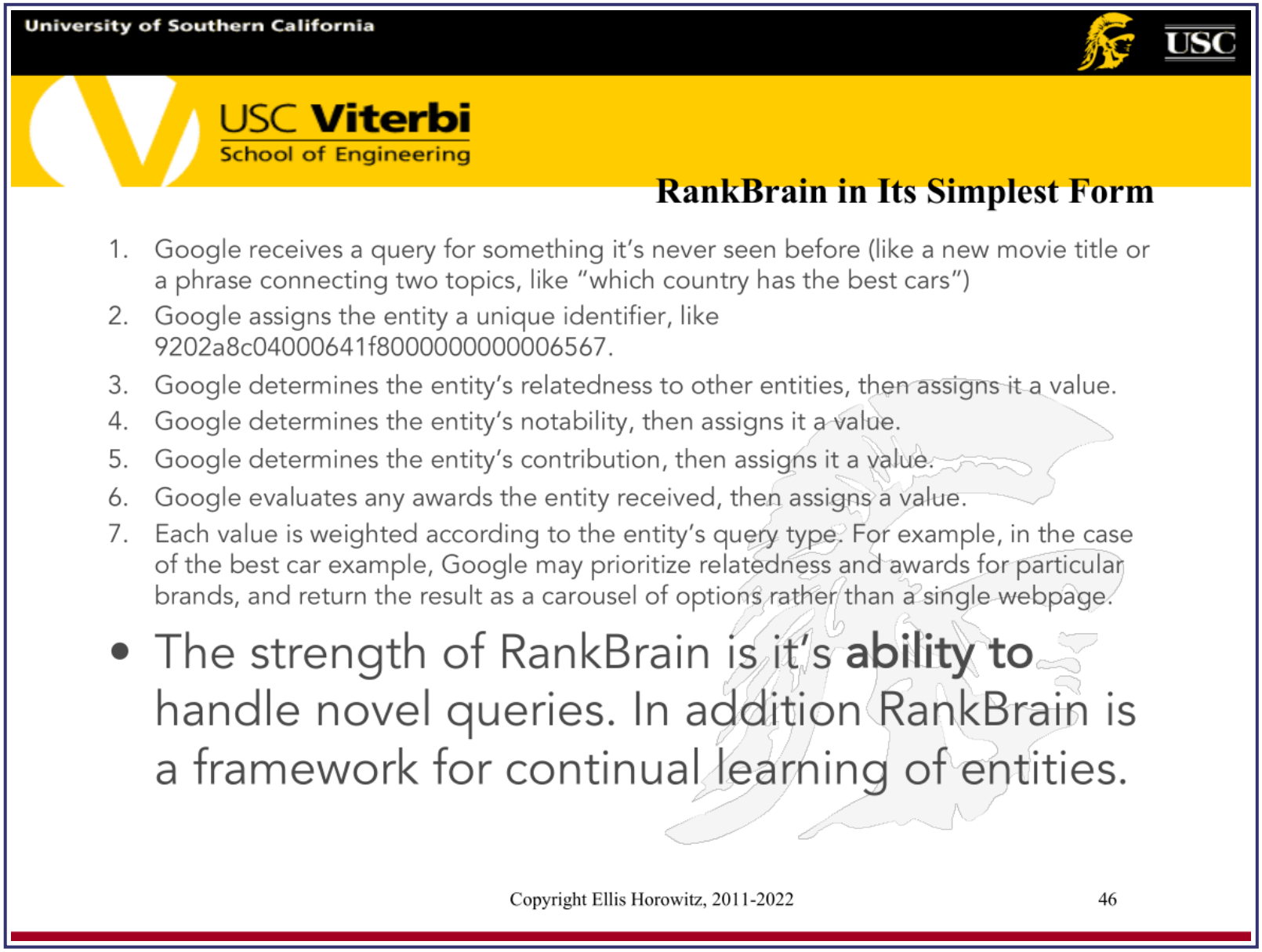
USC Viterbi School of Engineering

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)



●●


University of Southern California
USC
USC Viterbi
School of Engineering
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
46

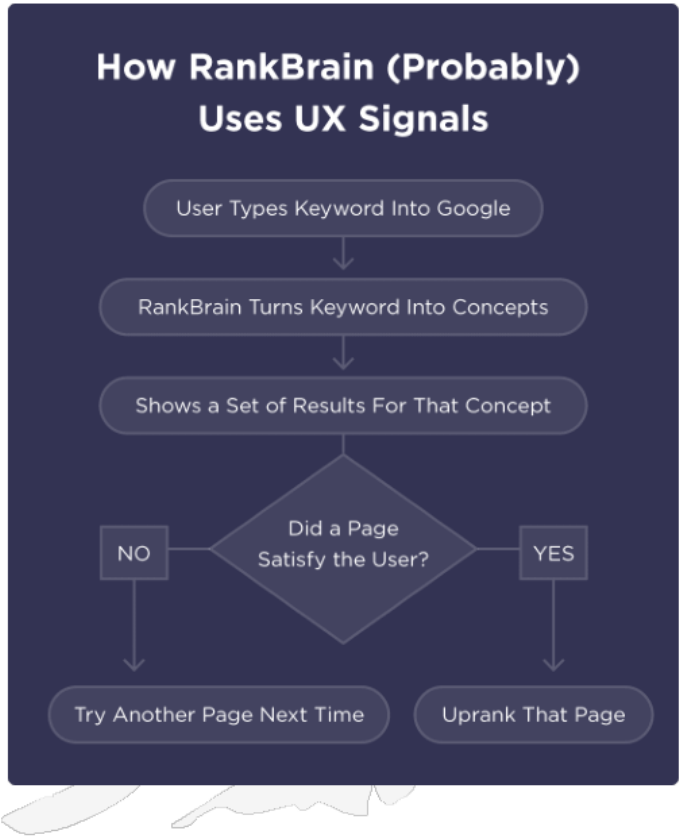
●●

University of Southern California

USC Viterbi School of Engineering

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).



●●

University of Southern California

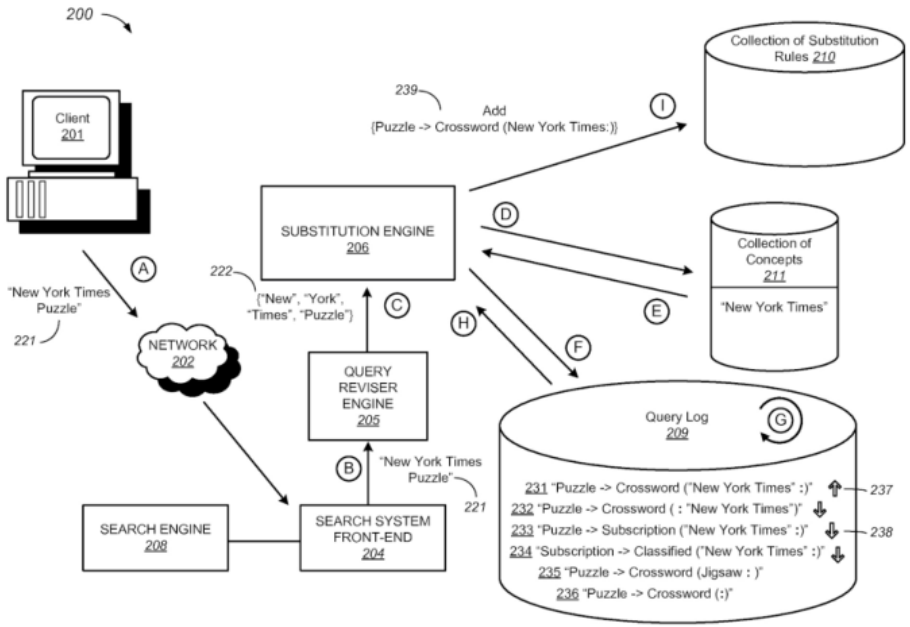
USC Viterbi School of Engineering

USC

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.



●●

University of Southern California

USC Viterbi School of Engineering

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

 50

••

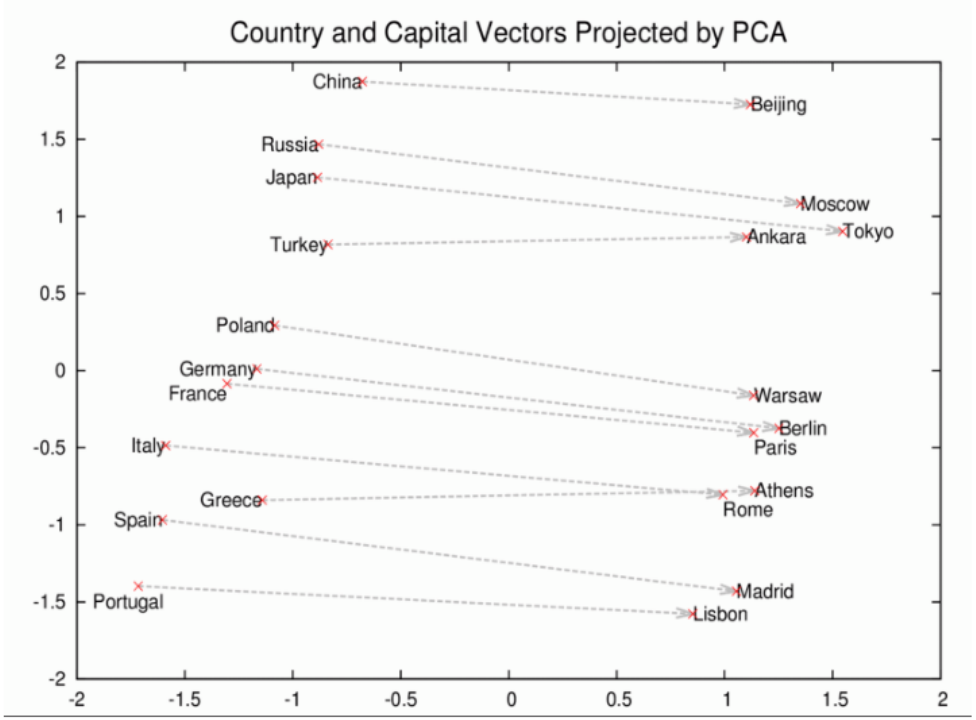
University of Southern California

USC Viterbi School of Engineering

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



 51