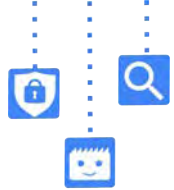
Improving Search Over the Years

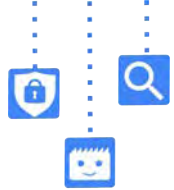
Paul Haahr
Distinguished Engineer, Search

Ranking Case Studies

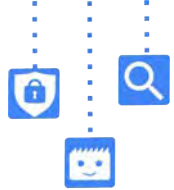
How do Google Search engineers think about ranking problems?



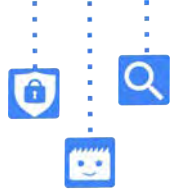
I could try to explain a step-by-step methodology...



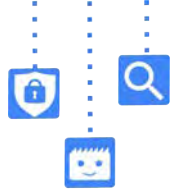
But there isn’t one.



There are many.



And they involve a lot of debugging, experimentation, evaluation, guesswork, research, and (often) luck.



Instead, here are some examples...

Synonyms and Siblings

Google's Synonyms System

- User vocabulary ≠ Document vocabulary
- System tries to bridge the gap by automatically adding alternative words
- Similar to using OR, but usually less important than original terms
- One of Google Search's most important ranking components

An example...

[cycling tours in italy]

⇒

[cycling OR cycle OR bicycle OR bike OR biking
tours OR tour OR holidays OR vacation
in
italy OR italian**]**

Contextual: Synonyms depend on other query words

[**gm** truck] ⇒ “general motors”

[**gm** barley] ⇒ “genetically modified”

[baseball **gm** salary] ⇒ “general manager”

Not the same as English Synonyms

- Designed to find good search results
- Hidden behind the scenes (mostly)
- Unimportant whether they're actually synonyms to a human reader

But...

For a short time in 2005, Google's top result for **[united airlines]** was continental.com

(The two companies did merge in 2010, but it wasn't our fault.)

(We hope.)

Why? Synonyms

(And a couple of unrelated bugs that I'm not going to talk about.)

[united airlines]

⇒

[united OR continental **airlines** OR air OR airline**]**

How do we fix things?

- We want algorithmic solutions
- Don’t just manually block the problems
- Look for patterns of failures

Synonyms sometimes finds siblings

- We can learn pairs of words that serve similar roles but aren't interchangeable
- Consider pairs of searches:

  [united *reservations*]
  [continental *reservations*]

  [united *newark airport*]
  [continental *newark airport*]

  ...

- "Siblings" (often rival siblings!)

Can we distinguish siblings from useful synonyms?

- Again, look to searches people do
- People compare siblings to each other:

```
[united vs continental]
[canon vs nikon]
[beatles vs stones]
[godzilla vs king kong]
...
```

- Look for `[X vs Y]` queries from logs, use as a negative signal for `X⇒Y` and `Y⇒X` synonyms

Then comes the hard part

- Process logs, build data, run experiments, evaluate the results, tune, repeat...
- Eventually, find many other synonym failures:

```
cat⇒dog
part time⇒full time
```

- But we also lost some good synonyms:

```
sign in⇒sign on
address⇒contact
```

Lessons

Understanding patterns of failures can reveal solutions

By not patching over algorithmic problems manually, we get more general solutions

Every change has wins and losses

Non-Compositional Compounds

Information Retrieval

Information Retrieval is mostly about matching and counting words

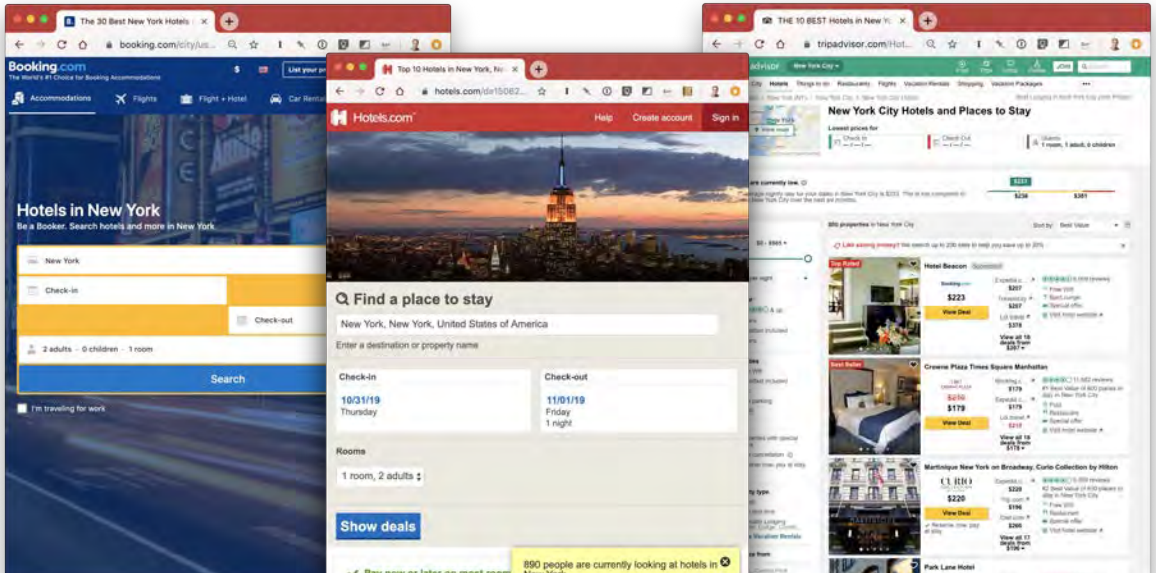
- Including title vs body, links, frequency, etc

This is the basic underpinning of Search.

Relevance comes from matching words

Consider these pages, which are good matches for **`[new york hotels]`**

- Title, body, links, etc



But sometimes it's a bit too simplistic

Are they good matches for `[york hotels]`?

Compounds

A compositional compound is “a phrase of two or more words where the words composing the phrase have the same meanings in the compound as their conventional meanings”.

A non-compositional compound is one where the meanings differ.

New York

"New York" is non-compositional.
Even though it is formed by compounding "New" and "York," there's nothing York-related now.

Not all place names follow the same rule

- "York" is not "New York"
- "Vegas" is "Las Vegas"

Can we identify non-compositional compounds?

Algorithm:

- Start with a set of “X Y” phrases
- Look at pages where “X Y” occurs
- If “X” or “Y” only appears in “X Y” on most of those pages, guess that it’s non-compositional

For “new york”:

- “new” appears alone on many
- “york” appears alone on very few

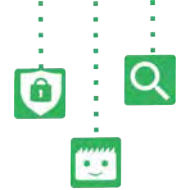
Matching NCCs

Now that we have non-compositional compounds, what do we do?

Specialized matching code:

```
[york hotels]
⇒
[(ignore_left:new york) hotels]
```

- Meaning: “Don’t match ‘york’ if the word to the left is ‘new’.”



Lessons

Edge case: would be very hard to predict in advance, but obvious to the first person who tries this query

Once seen, it's obvious there is a general pattern here

`[fantasy game]` is not "final fantasy"

`[view office]` is not "mountain view"

...

Hard work is done offline, ahead of time

Small change in matching code

Webmaster
Conference
Google

Language Evolves Over Time

If you received “  ” in a text in 1996, when Google launched at Stanford, would you have known it meant “rolling on the floor, laughing”?

Probably not. The first emoji appeared in 1997.

Why Search for Emoji?

- People use emoji all the time
- But, often, they're not sure what they mean exactly
- So, they search for them

Emoji in Search

Unfortunately, for a long time, Search ignored emoji and other “special characters”

- “Nobody searches for them”
- Expensive to index if they’re not used

What happened?

[😀] didn’t find anything

[**smiley face** 😀] sort-of worked

[😀 **meaning**] found dictionaries

Index/Query Alignment

Changes to what Search indexes are complicated, because they need to go in the right order:

- First, update indexing to allow emoji
- Wait for documents to be reindexed
- Then, change query parsing

But first:

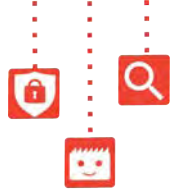
- Prove the cost is worth it!
- Even though they didn't work, people were using emoji in >1 million searches per day

But...

After data was in the index and query parsing was fixed, we ran evaluations, with very negative results.

Lots of other systems and models that needed to be updated before launch:

- Link processing
- Spelling
- Autocomplete
- ...



While we were at it...

Also added math and other symbols

- [∞]
- [∑x]
- [P≠NP]

And someone else did (some) punctuation

- [+=]
- [== vs ===]
- [P!= NP]

Lessons

Things which look easy
from the outside can be
a lot of work to implement

All the assumptions
you bake into your code
can change over 20 years

The Search Emoji Team