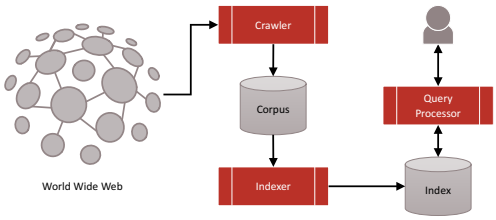
UFMG UNIVERSIDADE FEDERAL DE MINAS GERAIS

Ranking Models

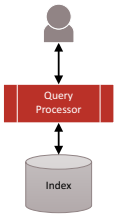
Query Understanding

Rodrygo L. T. Santos
rodrygo@dcc.ufmg.br

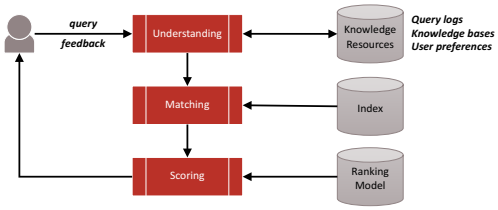
Search components



Search components



Query processing overview



Query processing overview

Queries and information needs

A query can represent very different information needs
- May require different search techniques and ranking algorithms to produce the best rankings

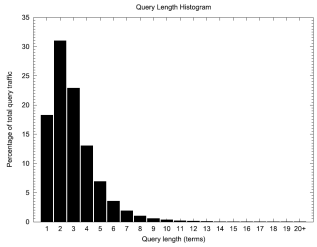
A query is often a poor representation of a need
- Users may find it difficult to express what they want
- Users may assume the search engine will guess

Complexity matters

Shorter
More popular
Less complex

Longer
Less popular
More complex

Query length



Power law distribution:
- $p(k) = Ck^{-s}$ for $k \geq k_o$
- C is normalizing constant, s is slope, k_0 is lower bound from which power law holds

Long queries

Yahoo! (2006) claimed 17% queries with 5+ words
- Current trend toward longer queries

Task-oriented search
- Question answering, literature search, cut-and-paste

Voice-activated search
- Cortana, Siri, Google Now

Complex queries

Long queries are also complex
- Rarity of verbose queries
- High degree of query specificity
- Term redundancy or extraneous terms (lot of noise)
- Lack of sufficient natural language parsing
- Hard to distinguish key and complementary concepts

Context matters

"It's raining"
- ... says the weatherman, conveying the weather
- ... writes the poet, conveying sadness in their work
- ... says your mom, indicating you should put on a coat
- ... says one bored person to another

Query understanding

About what happens before ranking
- How users express their queries
- How we can interpret their needs

Queries as first-class citizens
- ~~How to improve ranking regardless of query~~
- How to improve query regardless of ranking

A host of techniques

Query preprocessing
- Language detection
- Character filtering
- Query tokenization
- Spelling correction
- Inflection handling

Query rewriting
- Query relaxation
- Query expansion
- Query segmentation
- Query scoping

Spelling correction

10-15% of all web queries have spelling errors
- For today's searchers, a search engine without robust spelling correction simply doesn't work

How to (mis)spell "Britney Spears"

- britney spears
- brittany spears
- brittney spears
- britany spears
- britny spears
- briteny spears
- britteny spears
- briney spears
- brittny spears
- brintey spears
- britanny spears
- britiny spears
- britnet spears
- britiney spears
- britaney spears
- britnay spears
- brithney spears
- brtiney spears
- birtney spears

…

Spelling correction

Identify misspelled query words
- Those not found in a spelling dictionary

Identify candidate corrections
- Dictionary words similar to the misspelled word

Display candidate corrections
- Ideally, the single best one

Identifying candidate corrections

Compute edit distance
- Minimum number of insertions, deletions, substitutions, or transpositions of single characters

extenssions → extensions *(insertion error)*
poiner → pointer *(deletion error)*
marshmellow → marshmallow *(substitution error)*
brimingham → birmingham *(transposition error)*

Identifying candidate corrections

Edit distance calculation can be sped up
- Restrict to words starting with same character
- Restrict to words of same or similar length
- Restrict to words that sound the same

Phonetic encoding (Soundex)

1. Keep the 1st letter (in uppercase)
2. Replace with hyphens:
   a, e, i, o, u, y, h, w → –
3. Replace with numbers:
   b, f, p, v → 1 *l* → 4
   c, g, j, k, q, s, x, z → 2 *d, t* → 3
   m, n → 5 *r* → 6
4. Delete adjacent repeats of a number
5. Delete hyphens
6. Keep first 3 numbers and pad with zeros

extenssions	extensions
1. Extenssions	1. Extensions
2. Ext–nss—ns	2. Ext–ns—ns
3. E23–522—52	3. E23–52—52
4. E23–52—52	4. E23–52—52
5. E235252	5. E235252
6. E235	6. E235

Phonetic encoding (Soundex)

1. Keep the 1st letter (in uppercase)
2. Replace with hyphens:
   a, e, i, o, u, y, h, w → –
3. Replace with numbers:
   b, f, p, v → 1 *l* → 4
   c, g, j, k, q, s, x, z → 2 *d, t* → 3
   m, n → 5 *r* → 6
4. Delete adjacent repeats of a number
5. Delete hyphens
6. Keep first 3 numbers and pad with zeros

poiner	pointer
1. Poiner	1. Pointer
2. P—n–r	2. P—nt–r
3. P—5–6	3. P—53–6
4. P—5–6	4. P—53–6
5. P56	5. P536
6. P560	6. P536

Displaying the best correction

There might be several candidate corrections
- We can display only one ("Did you mean …")

Best correction depends on context
- lawers → lowers, lawyers, layers, lasers, lagers
- trial lawers → trial lawyers

Could mine query logs or other corpora for stats

Handling word inflections

Option #1
- **Stem** both documents and query
  [rock climbing] → [rock climb]

Option #2
- **Expand** query with inflection variants
  [rock climbing] → [rock {climbing climb}]

Query-based stemming

Delay stemming until we see a query
- Improved flexibility, effectiveness

Leverage context from surrounding words
- [logistic manager] → [{logistic logistics} manager]
- [logistic regression] → [logistic regression]

Stem classes

Stem classes identified by stemming large corpora

bank: { bank banked banking bankings banks }
ocean: { ocean oceaneering oceanic oceanics oceanization oceans }
polic: { polic polical polically police policeable policed policement policer policers polices policial policically policier policiers … }

Often too big and inaccurate
- Modify using analysis of word co-occurrence

Query rewriting

Rewriting for recall
- Query relaxation
- Query expansion

Rewriting for precision
- Query segmentation
- Query scoping

Query rewriting for recall

Some queries may return very limited sets of results
- Some may return nothing (aka *null queries*)

Vocabulary mismatch problem
- Searcher and publisher's vocabularies may differ

Solution: bridge the gap by tuning query specificity
- Either remove or add terms as required

Query relaxation

Rather than a verbose query, fire a shorter version!
- [ideas for breakfast menu for a staff meeting]
  ↳ [breakfast meeting menu ideas]
- [Provide information on international support provided to either side in the Spanish Civil War]
  ↳ [spanish civil war]

Query relaxation approaches

How to discard useless (or keep useful) terms?
- Several feature-based machine learning approaches (classification, regression, clustering)

Key considerations
- How to identify sub-query candidates?
- What features best describe a sub-query?

Identifying sub-query candidates

Individual words

Sequences of 2+ words

Combinations of 2+ words

Salient phrases (noun phrases, named entities)

Right part of the query

Sub-query features

Frequency statistics (TF, MI) in multiple corpora
- Google n-grams, Wiki titles, query logs

Linguistic features
- POS tags, entities, acronyms, stopwords

Sub-query features
- Length, category, similarity/position wrt query

Query expansion

Bridge vocabulary mismatch with added words

- Adding alternative words
  [vp marketing] → [(vp OR vice president) marketing]
  [laptop repair] → [(laptop OR computer) repair]
- Adding related words
  [tropical fish] → [tropical fish aquarium exotic]

Alternative words expansion

Acronyms matched in dictionaries

VP: Vice President
VP: Vice Principal

Acronyms mined from text

Business intelligence (BI) *combines a broad set of data analysis applications, including* ***online analytical processing (OLAP)****, and* ***data warehousing (DW)****.*

Alternative words expansion

Synonyms matched in dictionaries

laptop: computer
laptop: notebook

Synonyms mined via similar contexts

- Cosine of word embeddings (e.g., word2vec)
  (see ***Latent Semantic Models*** class)

Related words expansion

Relatedness via word co-occurrence

- Either in the entire document collection, a large collection of queries, or the top-ranked documents

Several co-occurrence measures

- Mutual information, Pearson's Chi-squared, Dice

Interactive query expansion

Require user's (explicit, implicit) feedback

- Rated, clicked, viewed documents
  (see ***Feedback Models*** class)

Query rewriting for precision

Query relaxation and expansion improve recall

- Avoid small or empty result sets

We also want to improve precision

- Avoid large and noisy result sets

Solution: improve the focus of the query

- Identify key segments and scopes

Query segmentation

Queries often contain multiple semantic units
- [new battery charger for hp pavilion notebook]
  ↳ [new battery charger hp pavilion notebook]

Leverage query structure via segmentation
- Identify multiple segments
- Process segments separately
  (see ***Structural Models*** class)

Query segmentation

A query with n tokens has $n-1$ split points
- We can have a total of 2^{n-1} possible segmentations

How to find the best segmentation?

[machine learning toolkit]
[machine learning toolkit]
[machine learning toolkit]
[machine learning toolkit]

Query segmentation approaches

Several approaches
- Dictionary-based approaches
- Statistical approaches
- Machine-learned approaches

Dictionary-based segmentation

Simplest approach
- A segment is a phrase in a dictionary

Drawback #1: dictionary coverage
- e.g., machine learning not found

Drawback #2: segment overlap
- e.g., both machine learning and learning toolkit found

Statistical segmentation

Exploits word collocations
- A word is in a segment if it co-occurs with the other words already in the segment above a threshold

Drawback: threshold sensitivity
- Threshold determines a trade-off (precision vs. recall)
- Threshold is corpus and language specific

Machine-learned segmentation

A binary classification approach
- Each token either continues a segment or not

Tokens represented as feature vectors
- e.g., token frequency, mutual information, POS tags

Drawback: data labeling for training
- Must manually segment lots of queries

Query scoping

Add a tag to each query segment
- Attributes in structured domains
  [black michael kors dress]
  ↳ [black:color michael kors:brand dress:category]
- Semantic annotations in open domains
  [microsoft ceo]
  ↳ [microsoft:company-3467 ceo:occupation-7234]

Tagging query segments

Segment tagging as non-binary, sequential prediction
- Classes known in advance (e.g., document fields, product attributes, knowledge base entries)

Several approaches
- Dictionary-based approaches
- Graphical modeling approaches

Exploiting tagged scopes

Attribute scoping
- Match each segment against its tagged attribute

Semantic scoping
- Promote semantically related matches (e.g., documents with entities close to the query entity)

Summary

Users provide limited evidence of their needs
- And yet expect fantastic search results

Query understanding helps bridge the gap
- Better recall through relaxation and expansion
- Better precision through segmentation and scoping

Open up possibilities for effective ranking!

References

Information Retrieval with Verbose Queries
Gupta and Bendersky, FnTIR 2015

Search Engines: Information Retrieval in Practice, Ch. 6
Croft et al., 2009

Introduction to Information Retrieval, Ch. 3
Manning et al., 2008

References

Query Understanding
Tunkelang, 2017

UFMG UNIVERSIDADE FEDERAL DE MINAS GERAIS

Coming next…

Vector Space Models

Rodrygo L. T. Santos
rodrygo@dcc.ufmg.br