Chapter 6

Queries and Interfaces
Keyword Queries

- Simple, natural language queries were designed to enable everyone to search.
- Current search engines do not perform well (in general) with natural language queries.
- People trained (in effect) to use keywords:
  - Compare average of about 2.8 words/Web query to average of 30 words/Community-based Question Answering (CQA) query.
- Keyword selection is not always easy:
  - Query refinement techniques can help.
Stem Classes

- Stemming generates *stem classes*

- A *stem class* is the group of words that will be transformed into the *same stem* by the stemming algorithm
  - Generated by running *stemmer* on large corpus
  - e.g., Porter stemmer on TREC News on 3 stem classes with the first entry being the *stem*

```
/bank banked banking bankings banks
/ocean oceaneering oceanic oceanics oceanization oceans
/polic polical polically policeable policed
-policement policer policers polices policial
-policically policier policiers policies policing
-policization policize policly policy policying policies
```
Stem Classes

- Stem classes are often too big and inaccurate
- Modify using analysis of word co-occurrence
- Assumption:
  - Word variants that could substitute for each other should co-occur often in documents. For example,
    - Meeting ~ Assembly
    - Adult ~ Grown up
Modifying Stem Classes

- For all pairs of words in the stem classes, count how often they co-occur in text windows of $W$ words, where $W$ is in the range 50-100

- Compute a co-occurrence or association metric for each pair, which measures the degree of association between the words

- Construct a graph where the vertices represent words and the edges are between words whose co-occurrence metric is above a threshold value, which is set empirically

- Find the connected components of the graph. These are the new stem classes.
Modifying Stem Classes

- Dices’ Coefficient is an example of a term association measure
  - \(2 \frac{n_{ab}}{(n_a + n_b)}\), where \(n_x\) is the number of windows (documents) containing \(x\)
  - a measure of the proportion of term co-occurrence
- Two vertices are in the same connected component of a graph if there is a path between them
  - Forms word clusters based on a threshold value
- Sample output of modification
  
  ```
policies policy
/police policed policing
/bank banking banks
```
Spell Checking

- Important part of *query processing*
  - 10-15% of all Web queries have *spelling errors*
- There are many types of errors, e.g., errors extracted from query logs

poiner sisters
brimingham news
catamarn sailing
hair extenssions
marshmellow world
miniture golf courses
psyhics
home doceration
realstateisting.bc.com
akia 1080i manunal
ultimatwarcade
mainscourcebank
dellottitouche
Spell Checking

- Basic approach: *suggest corrections* for words not found in *spelling dictionary*
  - Many spelling errors are related to websites, products, companies, people, etc. that are unlikely to be found

- Suggestions found by comparing word to words in dictionary using *similarity measure*

- Most common similarity measure is *edit distance*
  - Number of operations required to transform one word into the other
Edit Distance

- **Damerau-Levenshtein Distance**
  - Counts the minimum number of *insertions, deletions, substitutions, or transpositions* of single characters required
  - e.g., Damerau-Levenshtein distance 1 (single-character errors)

  - extensions → extensions (insertion error)
  - poineer → pointer (deletion error)
  - marshmellow → marshmallow (substitution error)
  - brimingham → birmingham (transposition error)

- **Distance 2**

  - doceration → deceration
  - deceration → decoration
Edit Distance

- Different techniques used to *speed up* calculation of edit distances -- restrict to words that
  - start with *same character* (spelling errors rarely occur in the first letter)
  - come with similar *length* (spelling errors rarely occur on words with the same length)
  - *sound* the *same* (*homophone*, rules map words to codes)

- Last option uses a (same) *phonetic code* to group words
  - e.g., Soundex, a *phonetic index* grouping words that *sound alike* but are *spelled differently*
Soundex Code

1. Keep the first letter (in upper case).

2. Replace these letters with hyphens: a,e,i,o,u,y,h,w.

3. Replace the other letters by numbers as follows:
   1: b,f,p,v
   2: c,g,j,k,q,s,x,z
   3: d,t
   4: l
   5: m,n
   6: r

4. Delete adjacent repeats of a number.

5. Delete the hyphens.

6. Keep the first three numbers or pad out with zeros.

extentssions → E235; extensions → E235
marshmellow → M625; marshmallow → M625
brimingham → B655; birmingham → B655
poiner → P560; pointer → P536  (correct word may not always have the same Soundex code)
Spelling Correction Issues

- Ranking corrections (> 1 possible corrections for an error)
  - “Did you mean...” feature requires accurate ranking of possible corrections (more likely: the best suggestion)

- Context
  - Choosing right suggestion depends on context (other words)
  - e.g., lawers → lowers, lawyers, layers, lasers, lagers but trial lawers → trial lawyers

- Run-on errors (word boundaries are skipped/mistyped)
  - e.g., “mainscourcebank”
  - *Missing spaces* can be considered another single character error in right framework
Noisy Channel Model

- Address the issues of ranking, context, and run-on errors

- User chooses word $w$ based on probability distribution $P(w)$
  
  - Called the language model
  
  - Can capture context information about the frequency of occurrence of a word in text, e.g., $P(w_1 \mid w_2)$
  
  - The probability of observing a word, given that another one has just been observed

- User writes word $w$, but noisy channel causes word $e$ to be written instead with probability $P(e \mid w)$
  
  - Called error model
  
  - Represents information about the frequency of spelling errors
Noisy Channel Model

- Need to estimate *probability of correction* – to represent info. about the *frequency* of different types of *errors*

  \[ P(w | e) = P(e | w) P(w) \text{, i.e., the probability that given a written word } e, \text{ the correct word is } w \]

- Estimate *language model probability* using *context*

  \[ P(w) = \lambda P(w) + (1 - \lambda) P(w | wp) \]

  where *wp* is a *previous* word of *w*, and *\( \lambda \)* is a parameter which specifies the *relative importance* of \( P(w) \) & \( P(w | wp) \)

- **Examples.**

  - “fish tink”: “tank” and “think” both likely corrections, but \( P(\text{tank} \mid \text{fish}) > P(\text{think} \mid \text{fish}) \)
Noisy Channel Model

- Language model probabilities estimated using corpus and query log

- Both simple and complex methods have been used for estimating error model
  - Simple approach: assume that all words with same edit distance have same probability, only edit distance 1 and 2 considered
  - More complex approach: incorporate estimates based on common typing errors
    - Estimates are derived from large collections of text by finding many pairs of (in)correctly spelled words
Relevance Feedback

- A *query expansion* and *refinement* technique

- User *identifies relevant* (and maybe non-relevant) documents in the initial result list

- System modifies query using terms from those documents and *re-ranks* documents

- **Pseudo-relevance feedback**
  - Assumes top-ranked documents are relevant – no user input
  - Keywords are added/dropped or their weights increase/decrease
Pseudo-relevance Feedback Example

1. **Badmans Tropical Fish**
   A freshwater aquarium page covering all aspects of the tropical fish hobby. ... to Badman's Tropical Fish. ... world of aquariology with Badman's Tropical Fish. ...

2. **Tropical Fish**
   Notes on a few species and a gallery of photos of African cichlids.

3. **The Tropical Tank Homepage - Tropical Fish and Aquariums**
   Info on tropical fish and tropical aquariums, large fish species index with ... Here you will find lots of information on Tropical Fish and Aquariums. ...

4. **Tropical Fish Centre**
   Offers a range of aquarium products, advice on choosing species, feeding, and health care, and a discussion board.

5. **Tropical fish - Wikipedia, the free encyclopedia**
   Tropical fish are popular aquarium fish, due to their often bright coloration. ... Practical Fishkeeping • Tropical Fish Hobbyist • Koi. Aquarium related companies: ...

6. **Tropical Fish Find**
   Home page for Tropical Fish Internet Directory ... stores, forums, clubs, fish facts, tropical fish compatibility and aquarium ...

7. **Breeding tropical fish**
   ... interested in keeping and/or breeding Tropical, Marine, Pond and Coldwater fish. ... Breeding Tropical Fish ... breeding tropical, marine, coldwater & pond fish ...

8. **FishLore**
   Includes tropical freshwater aquarium how-to guides, FAQs, fish profiles, articles, and forums.

9. **Cathy's Tropical Fish Keeping**
   Information on setting up and maintaining a successful freshwater aquarium.

10. **Tropical Fish Place**
    Tropical Fish information for your freshwater fish tank ... great amount of information about a great hobby, a freshwater tropical fish tank. ...
Relevance Feedback Example

- If we assume top 10 are relevant, most frequent terms are (with frequency):
  - a (926), td (535), href (495), http (357), width (345), com (343), nbsp (316), www (260), tr (239), htm (233), class (225), jpg (221)

  - Too many stopwords and HTML expressions

- For query expansion, use only snippets and remove stopwords
  - tropical (26), fish (28), aquarium (8), freshwater (5), breeding (4), information (3), species (3), tank (2), Badman’s (2), page (2), hobby (2), forums (2)
Relevance Feedback - Query Logs

- Drawback of the pseudo-relevance feedback strategy:
  - When the initial ranking does not contain many relevant documents, the expansion are unlikely to be helpful

- *Query logs* provide important contextual information that can be used effectively for query expansion

- Context in this case is
  - Previous queries that are the *same*
  - Previous queries that are *similar*
  - Query sessions including the *same query*

- Query history for individuals could be used for caching
Relevance Feedback

- **Rocchio algorithm**
  - Based on the concept of *optimal query*
  - **Maximizes** the difference between the
    1. *average* vector representing the *relevant* documents, and
    2. *average* vector representing the *non-relevant* documents

- **Modifies** query according to

\[ q'_j = \alpha q_j + \beta \frac{1}{|\text{Rel}|} \sum_{D_i \in \text{Rel}} d_{ij} - \gamma \frac{1}{|\text{Nonrel}|} \sum_{D_i \in \text{Nonrel}} d_{ij} \]

- \( \alpha, \beta, \) and \( \gamma \) are parameters
  - Typical values 8, 16, and 4
Snippet Generation

- Successful search engine interface depends on users’ understanding of the (contents of) query results

Tropical Fish
One of the U.K.s Leading suppliers of Tropical, Coldwater, Marine Fish and Invertebrates plus... next day fish delivery service...
www.tropicalfish.org.uk/tropical_fish.htm  Cached page

- Snippets are query-dependent document summaries

- Snippet generation is a simple text summarization
  - Rank each sentence in a document using a significance factor, first proposed by H. P. Luhn in 50’s
  - Select the top sentences for the summary with a number of significance words
Sentence Selection

- Significance factor for a sentence is calculated based on the occurrence of significant words
  - If $f_{d,w}$ is the frequency of word $w$ in document $d$, then $w$ is a significant word if it is not a *stopword*, i.e., a high-frequency word, and
    
    \[
    f_{d,w} \geq \begin{cases} 
    7 - 0.1 \times (25 - s_d), & \text{if } s_d < 25 \\
    7, & \text{if } 25 \leq s_d \leq 40 \\
    7 + 0.1 \times (s_d - 40), & \text{otherwise}
    \end{cases}
    \]

    where $s_d$ is the *number of sentences* in document $d$

  - Text is *bracketed* by significant words (limit on number of non-significant words in bracket)
Sentence Selection

- **Significance factor** for bracketed text spans is computed by (i) dividing the *square* of the number of *significant* words in the span by (ii) the *total* number of words.

- **Example.**

  \[
  \text{Significance factor} = \frac{4^2}{7} = 2.3
  \]

  \[
  \text{Example sentence: } w \ w \ w \ w \ w \ w \ w \ w \ w \ w \ w \ w \ w \ w \ w \ w \ w. \\
  \text{(Initial sentence)}
  \]

  \[
  \text{Significant words: } w \ w \ s \ w \ s \ s \ w \ w \ s \ w \ w. \\
  \text{(Identify significant words)}
  \]

  \[
  \text{Bracketed span: } w \ w \ [s \ w \ s \ s \ w \ w \ s \ w \ s] \ w \ w. \\
  \text{(Text span bracketed by significant words)}
  \]

  The limit set for non-significant words in a bracket is typically 4.
Snippet Generation

- Involves more features than just *significance factor*

- A typical sentence-based, snippet-generation approach:
  1. Whether the *sentence* is a *heading*
  2. Whether it is the 1\textsuperscript{st} or 2\textsuperscript{nd} line of the document
  3. The total number of *query terms* occurring in the *sentence*
  4. The number of *unique query terms* in the *sentence*
  5. The longest contiguous run of query words in the *sentence*
  6. A *density* measure of query words (i.e., *significance factor* on query words in sentences)

- *Weighted* combination of *features* used to rank *sentences*