Roadmap

- Query types and operations
  - Forms of text-retrieval queries
  - Relevance feedback,
  - Query expansion
  - Query reformulation
  - Allowing Errors/typos in queries
- Web retrieval
  - Web spiders / Crawlers
  - Search engines
Forms of text-retrieval queries
IR queries

- Keyword queries
  - Covered so far
- Boolean queries (using AND, OR, NOT)
- Phrase queries
- Proximity queries
- Pattern Matching
IR queries: Phrase queries

- **Phrase**: ordered list of contiguous words.
- We want to be able to answer queries such as “Stanford university” – as a phrase
  - Thus the sentence “I went to university at Stanford” is not a match.
  - The concept of phrase queries has proven easily understood by users;
  - One of the few “advanced search” ideas that works
May allow *intervening stop words* and/or *stemming*.

- “buy camera” matches: “buy a camera”
- “buying the cameras”
IR queries: Phrase queries

- For this, it no longer suffices to store only \(<\text{term : docs}>\) entries
- Must have an inverted index that also stores \textit{positions of each keyword in a document}.

- Implementation:
  - Retrieve documents and positions for each individual word,
  - check for \textit{ordered contiguity of keyword positions}. 

Two or more separately matching term occurrences are within a specified distance.

- Distance is the number of intermediate words or characters.

In addition to proximity, some implementations may also impose a constraint on the word order.
Boolean syntax and operators

Sometimes query operators like NEAR, NOT NEAR, FOLLOWED BY, NOT FOLLOWED BY, SENTENCE or FAR are used to indicate a proximity-search limit between specified keywords:

for example,

"brick NEAR house".
Example:

- “dogs” and “race” within 4 words
  - match “…dogs will begin the race…”

May also perform **stemming** and/or **not count stop words**.

Implementation:

- Use approach similar to phrasal search to find documents in which all keywords are found in a context that satisfies the proximity constraints.
IR queries: Pattern Matching

- Allow queries that **match strings** rather than **word tokens**.
- Requires more **sophisticated data structures** and algorithms than inverted indices to retrieve efficiently.
IR queries: Pattern Matching

- **Prefixes**: Pattern that matches start of word.
  - “anti” matches “antiquity”, “antibody”, etc.

- **Suffixes**: Pattern that matches end of word:
  - “ix” matches “fix”, “matrix”, etc.

- **Substrings**: Pattern that matches arbitrary subsequence of characters.
  - “rapt” matches “enrapture”, “velociraptor” etc.

- **Ranges**: Pair of strings that matches any word lexicographically (alphabetically) between them.
  - “tin” to “tix” matches “tip”, “tire”, “title”, etc.
IR queries

- Full document queries
  - Example-based search
- Natural language questions
  - Where is my mom?
Query operations
Relevance Feedback

- After initial retrieval results are presented, allow the user to provide feedback on the relevance of one or more of the retrieved documents.
- Use this feedback information to reformulate the query.
- Produce new results based on reformulated query.
- Allows more interactive, multi-pass process.
  - Iterative process
Relevance Feedback

Query String

Revised Query

Query Reformulation

Feedback

Document corpus

IR System

Ranked Documents

ReRanked Documents

1. Doc1
2. Doc2
3. Doc3

1. Doc1
2. Doc2
3. Doc3

1. Doc2
2. Doc4
3. Doc5
Query Reformulation

- Revise query to account for feedback:
  - **Query Expansion**: Add new terms to query from relevant documents.
  - **Term Reweighting**: Increase weight of terms in relevant documents and decrease weight of terms in irrelevant documents.

- Several algorithms for query reformulation.
Query Reformulation in the Vector Model

- Change query vector using vector algebra.
- **Add** the vectors for the **relevant** documents to the query vector.
- **Subtract** the vectors for the **irrelevant** docs from the query vector.
- This both adds both positive and negatively weighted terms to the query as well as reweighting the initial terms.
Assume that the relevant set of documents $C_r$ are known.

Then the best query that ranks all and only the relevant queries at the top is: (Where $N$ is the total number of documents)

$$
\vec{q}_{opt} = \frac{1}{|C_r|} \sum_{\forall \vec{d}_j \in C_r} \vec{d}_j - \frac{1}{N - |C_r|} \sum_{\forall \vec{d}_j \notin C_r} \vec{d}_j
$$
Why is Feedback Not Widely Used

- Users sometimes reluctant (unwilling) to provide explicit feedback.
- Results in long queries that require more computation to retrieve, and search engines process lots of queries and allow little time for each one.
- Makes it harder to understand why a particular document was retrieved.
Why is Feedback Not Widely Used

- Users sometimes reluctant (unwilling) to provide explicit feedback.
- Results in long queries that require more computation to retrieve, and search engines process lots of queries and allow little time for each one.
- Makes it harder to understand why a particular document was retrieved.
Pseudo Feedback

- Use relevance feedback methods without explicit user input.
- Just assume the top $m$ retrieved documents are relevant, and use them to reformulate the query.
- Allows for query expansion that includes terms that are correlated with the query terms.
Query expansion

- Query can be automatically expanded with synonyms and related words from the thesaurus.
A thesaurus provides information on synonyms and semantically related words and phrases.

Example:

physician

   syn: ||croaker, doc, doctor, MD, medical, mediciner, medico, ||sawbones
   rel: medic, general practitioner, surgeon,
Query expansion

- Index-time vs Query-time expansion.

**Synonyms** for dog

**noun** canine mammal

- pup
- puppy
- hound
- mongrel
- stray
- tyke
- man's best friend
- fido
- flea bag
- bowwow
- tail-wagger
Why Query expansion?

- Query expansion to increase the quality of user search results.
- It is assumed that users do not always formulate search queries using the best terms.
- Also, the IR database may not contain the user entered terms.
Thesaurus-based Query Expansion

- For each term, $t$, in a query, expand the query with synonyms and related words of $t$ from the thesaurus.
- May weight added terms less than original query terms.
- Generally increases recall.
- May significantly decrease precision, particularly with ambiguous terms.
  - “interest rate” $\rightarrow$ “interest rate fascinate evaluate”
  - Context resolution
WordNet

- A more detailed database of semantic relationships between English words.
- Developed by famous cognitive psychologist George Miller and a team at Princeton University.
- About 144,000 English words.
- Nouns, adjectives, verbs, and adverbs grouped into about 109,000 synonym sets called synsets.
WordNet Synset Relationships

- **Antonym**: front → back
- **Attribute**: benevolence خير → good (noun to adjective)
- **Pertainym**: alphabetical → alphabet (adjective to noun)
- **Similar**: unquestioning → absolute
- **Cause**: kill → die
- **Entailment**: breathe تنفس → inhale استنشق
- **Holonym**: chapter → text (part to whole)
- **Meronym**: computer → cpu (whole to part)
- **Hyponym**: plant → tree (specialization)
- **Hypernym**: apple → fruit (generalization)
WordNet Query Expansion

- Add **synonyms** in the same synset.
- Add **hyponyms** to add specialized terms.
- Add **hypernyms** to generalize a query.
- Add **other related terms** to expand query.
Existing human-developed thesauri are not easily available in all languages.

Human thesauri are limited in the type and range of synonymy and semantic relations they represent.

Semantically related terms can be discovered from statistical analysis of corpora.

- Global analysis
- Local analysis
Determine term similarity through a pre-computed statistical analysis of the complete corpus.

Compute association matrices which quantify term correlations in terms of how frequently they co-occur.

Expand queries with statistically most similar terms.
Problems with Global Analysis

- **Term ambiguity** may introduce irrelevant statistically correlated terms.
  - “Apple computer” → “Apple red fruit computer”
- Since terms are highly correlated anyway, expansion may not retrieve many additional documents.
At query time, dynamically determine similar terms based on analysis of top-ranked retrieved documents.

Base correlation analysis on only the “local” set of retrieved documents for a specific query.

Avoids ambiguity by determining similar (correlated) terms only within relevant documents.

“Apple computer” →
“Apple computer Powerbook laptop”
Global vs. Local Analysis

- Global analysis requires intensive term correlation computation only once at system development time.

- Local analysis requires intensive term correlation computation for every query at run time (although number of terms and documents is less than in global analysis).

- But local analysis gives better results.
Query Expansion Conclusions

- Expansion of queries with related terms can improve performance, particularly recall.

- However, must select similar terms very carefully to avoid problems, such as loss of precision.
Allowing Errors
Allowing Errors

- What if query or document contains *typos* or *misspellings*?
- Judge similarity of words (or arbitrary strings) using:
  - *Edit distance* (Levenstein distance)
  - *Longest Common Subsequence* (LCS)
- Allow proximity search with bound on string similarity.
Allowing Errors: Edit (Levenstein) Distance

- **Edit (Levenstein) Distance**: The minimum number of character deletions, additions, or replacements needed to make two strings equivalent.
  - “misspell” to “mispell” is distance 1
  - “misspell” to “mistell” is distance 2
  - “misspell” to “misspelling” is distance 3

- Can be computed efficiently using *dynamic programming* in $O(mn)$ time
  - where $m$ and $n$ are the lengths of the two strings being compared.
Allowing Errors: Longest Common Subsequence (LCS)

- Length of the longest subsequence of characters shared by two strings.
- A subsequence of a string is obtained by deleting zero or more characters.

- Examples:
  - “misspell” to “mispell” is 7
  - “misspelled” to “misinterpreted” is 7
    “mis...p...e...ed”
Searching for Similar Words

- When spell-correcting a word, it is inefficient to serially search every word in the dictionary, compute the edit distance or LCS for each, and then take the most similar word.

- Use indexing to find most similar dictionary word without doing a linear search.
An inverted index for sequences of \( k \) characters contained in a word.

- \( 3 \)-grams for “index”: $in, ind, nde, dex, ex$.
  (where $ is a special char denoting start or end of a word)

For each \( k \)-gram encountered in the dictionary, the \( k \)-gram index has a pointer to all words that contain that \( k \)-gram.

- \( dex \rightarrow \{index, dexterity, ambidextrous\} \)
Using a $k$-gram Index

- Given a word, generate its “bag of $k$-grams” and use the $k$-gram index like a normal inverted index to find a word that contains many of the same $k$-grams.

- Like normal document retrieval except:
  - words → $k$-grams
  - documents → words

- Example:
  - Query: endex →{$en, end, nde, dex, ex$}
  - Retrieval Result: 1) index, 2) ended, 3) endear….
  - Compute detailed score just for top retrievals and take final top-scoring candidate.
Search Engines @ Web Spiders

Web Crawlers
Many names ...

- Crawler
  - Spider
  - Robot (or bot)
  - Web agent
  - Wanderer, worm, ...
And famous instances:

- **googlebot**, **scooter** (for the AltaVista public search engine), **slurp** (for the Yahoo public search engine), **msnbot**, …
How do they work?
How do they work?
Motivation for crawlers

- **Support universal search engines** (Google, Yahoo, MSN/Windows Live, Ask, etc.)
  - Building corpora (corpuses)
- **Vertical (specialized) search engines**, e.g. news, shopping, papers, recipes, reviews, etc.
- **Business intelligence**: keep track of potential competitors, partners
- **Monitor** Web sites of interest.
- **Evil**: harvest emails for spamming, phishing…
- … Can you think of some others?…
A crawler within a search engine

![Diagram of a crawler within a search engine](image)

- **Web**
  - **googlebot**
- **Page repository**
- **Text & link analysis**
- **Query**
- **hits**
- **Text index**
- **PageRank**
- **Ranker**

Sulieman Bani-Ahmad
3 April 2016
Basic crawlers

- This is a **sequential** crawler
- **Seeds** can be any list of starting URLs
- Order of page visits is determined by **frontier** data structure
  - Depth-First, Breadth-First
- **Stop criterion** can be anything
Graph traversal (**BFS** or **DFS**?)

- **Breadth First Search**
  - Implemented with QUEUE (FIFO)
  - Finds pages along shortest paths
  - If we start with “good” pages, this keeps us close; may be other good stuff…
Graph traversal (BFS or DFS?)

- Depth First Search
  - Implemented with STACK (LIFO)
  - Wander away (“lost in cyberspace”)
Queue: a FIFO list (shift and push)

```perl
my @frontier = read_seeds($file);
while (@frontier && $tot < $max) {
    my $next_link = shift @frontier;
    my $page = fetch($next_link);
    add_to_index($page);
    my @links = extract_links($page, $next_link);
    push @frontier, process(@links);
}
```
Don’t want to fetch same page twice!
   - Keep lookup table (hash) of visited pages

The frontier grows very fast!
   - May need to prioritize for large crawls

Fetcher must be robust!
   - Don’t crash if download fails
   - Timeout mechanism
Google: Basics

- What happens when you “Google” for something?
Google Search Results

- Database Google Used
- Approximate # of hits
- Ads selected by Google based on your search terms

**Search terms are in bold**

- URL, size, date last crawled
- Cached link
- Pages like this one
Stemming

- Google stems “when appropriate”
- Includes plural, singular, past, present tense of words in search
  - Search: school librarian
  - Result: library, librarian, library’s, librarian’s
- Single word searches aren’t stemmed
What Google doesn’t search

- Common or Stop words are ignored (unless you ask not to)
- No official list from Google
- Use quotation marks for phrases
  - Example:
    - “public librarian” 234,000 (.4% of)
    - public librarian 58,600,000
  - Forces searches on stop words
  - Turns off stemming
Exclusion and synonym operators

- **“-” exclusion operator**
  - Search:
    - twins Minnesota  2,750,000
  - Eliminate undesired words
    - twins Minnesota –sports  1,300,000

- **“~” synonym operator**
  - ~guide
    - searches for: tutorial, manual, help, map, tips
Search within a site

- Search within a site
  - site:memory.loc.gov “dust bowl”
- Use Google as a search engine for a site
- Can ONLY use first part of URL
  - Omit “http:” and final “/”
- Example:
  - site:bau.edu.jo “information technology”
### More examples

<table>
<thead>
<tr>
<th>This Search</th>
<th>Finds Pages Containing...</th>
</tr>
</thead>
<tbody>
<tr>
<td>biking Italy</td>
<td>the words biking and Italy</td>
</tr>
<tr>
<td>recycle steel <strong>OR</strong> iron</td>
<td>information on <strong>recycling steel</strong> or <strong>recycling iron</strong></td>
</tr>
<tr>
<td>&quot;I have a dream&quot;</td>
<td>the <strong>exact phrase</strong> I have a dream</td>
</tr>
<tr>
<td>salsa <strong>-dance</strong></td>
<td>the word salsa <strong>but NOT</strong> the word dance</td>
</tr>
<tr>
<td>Louis &quot;I&quot; France</td>
<td>information about Louis the First (I), weeding out other kings of France</td>
</tr>
<tr>
<td>fortune-telling</td>
<td>all forms of the term, whether spelled as a single word, a phrase, or hyphenated</td>
</tr>
<tr>
<td>**define:**imbroglio</td>
<td>definitions of the word imbroglio from the Web</td>
</tr>
</tbody>
</table>
Local time

- Get the current time for any location.

![Google search for time in Amman, Jordan](image)
Quickly call up a mapped location.

Google map Cleveland, OH

About 39,400,000 results (0.51 seconds)

Street Map Of Cleveland Ohio – wow.com

Search for Street Map Of Cleveland Ohio Look Up Quick Results Now!
Weather

- Current: Mostly Cloudy
- Wind: SW at 20 mph
- Humidity: 88%
- Temperature:
  - Monday: 71°F (52°F)
  - Tuesday: 78°F (54°F)
  - Wednesday: 79°F (55°F)
  - Thursday: 83°F (58°F)

More information can be found at:
- 10 Day Weather Forecast for Davis, CA (95616) - weather.com
- Hour by Hour Weather Forecast for Davis, CA (95616) - weather.com
- Davis, California (95616) Conditions & Forecast: Weather Underground

Sulieman Bani-Ahmad  3 April 2016
Define

technology

1. The application of scientific knowledge for practical purposes, esp. in industry: "computer technology"; "recycling technologies".
2. Machinery and equipment developed from such scientific knowledge. More »

Technology - Wikipedia, the free encyclopedia
Technology is the making, usage and knowledge of tools, techniques, crafts, systems or methods of organization in order to solve a problem or serve some...

Information technology - Emerging technologies - Educational technology
en.wikipedia.org/wiki/Technology - Cached - Similar

Technology News - Computers, Internet, Invention and Innovation...
Jun 6, 2011 ... Find information about the latest advances in technology at CNN. CNN Technology news and video covers the internet, business and personal ...
www.cnn.com/TECH/ - Cached - Similar
How can Google be used as a calculator?
<table>
<thead>
<tr>
<th>Operator</th>
<th>Function</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Addition</td>
<td>[ 15.99 + 32.50 + 13.25 ]</td>
</tr>
<tr>
<td>−</td>
<td>Subtraction</td>
<td>[ 79 − 18 − 19 ]</td>
</tr>
<tr>
<td>*</td>
<td>Multiplication</td>
<td>[ 2 * 3 * 7 ]</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
<td>[ 378 / 9 ]</td>
</tr>
<tr>
<td>^ or **</td>
<td>Exponentiation (raise to a power of)</td>
<td>[ 4^10 ] or [ 4**10 ]</td>
</tr>
<tr>
<td>% of</td>
<td>Percent</td>
<td>[ 15% of 93.45 ]</td>
</tr>
<tr>
<td>mod or %</td>
<td>modulo (the remainder after division)</td>
<td>[ 15 mod 9 ] or [ 15 % 9 ]</td>
</tr>
<tr>
<td>the $n$th root of</td>
<td>calculates the $n$th root</td>
<td>[ 4th root of 16 ]; [ cube root of 109 ]; [ square root of 42 ] or [ sqrt(42) ]</td>
</tr>
</tbody>
</table>
And more ..

<table>
<thead>
<tr>
<th>Operator</th>
<th>Function</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>sin, cos, tan, sec, csc, cot, etc.</td>
<td>Trigonometric functions (arguments are assumed to be in radians)</td>
<td>[ cos(\pi/6) ]; [ cosine(\pi/6) ]</td>
</tr>
<tr>
<td>arcsin, arccos, arctan, arccsc, etc.</td>
<td>Inverse trigonometric functions</td>
<td>[ arccos(.5) ]</td>
</tr>
<tr>
<td>sinh, cosh, tanh, csch, arsinh, arccsch, Hyperbolic functions etc.</td>
<td></td>
<td>[ cosh(6) ]</td>
</tr>
<tr>
<td>ln</td>
<td>Logarithm base e</td>
<td>[ ln(16) ]</td>
</tr>
<tr>
<td>log</td>
<td>Logarithm base 10</td>
<td>[ log(16) ]</td>
</tr>
<tr>
<td>lg</td>
<td>Logarithm base 2</td>
<td>[ lg(16) ]</td>
</tr>
<tr>
<td>exp</td>
<td>Exponential function</td>
<td>[ exp(16) ]</td>
</tr>
<tr>
<td>!</td>
<td>Factorial</td>
<td>[ 5! ]</td>
</tr>
<tr>
<td>choose</td>
<td>x choose y calculates the number of ways of choosing a set of y elements from a set of x distinct elements</td>
<td>[ 5 choose 3 ]</td>
</tr>
</tbody>
</table>
Solve Conversions

10 kilometers = 6.21371192 miles
# Units of Measure and Conversions

<table>
<thead>
<tr>
<th>Type of Units</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Currency</td>
<td>Australian Dollars (AUD), British pounds (GBP), Euros, US Dollars (USD)</td>
</tr>
<tr>
<td>Mass</td>
<td>kilogram or kg, grams or g, grains, pounds or lbs, carats, stones, tons, tonnes</td>
</tr>
<tr>
<td>Length</td>
<td>meters or m, miles, feet, Angstroms, cubits, furlongs</td>
</tr>
<tr>
<td>Volume</td>
<td>gallons, liters or l, bushels, teaspoons, pints</td>
</tr>
<tr>
<td>Area</td>
<td>square kilometers, acres, hectares</td>
</tr>
<tr>
<td>Time</td>
<td>days, seconds or s, centuries, sidereal years, fortnights</td>
</tr>
<tr>
<td>Electricity</td>
<td>volts, amps, ohms, henrys</td>
</tr>
<tr>
<td>Energy</td>
<td>Calories, British thermal units (BTU), joules, ergs, foot-pounds</td>
</tr>
<tr>
<td>Power</td>
<td>watt, kilowatts, horsepower or hp</td>
</tr>
<tr>
<td>Information</td>
<td>bits, bytes, kbytes, etc.</td>
</tr>
<tr>
<td>Quantity</td>
<td>dozen, baker’s dozen, percent, gross, great gross, score, googol</td>
</tr>
<tr>
<td>Numbering systems</td>
<td>decimal, hexadecimal or hex, octal, binary, roman numerals, etc.</td>
</tr>
</tbody>
</table>

Prefix hexadecimal numbers with 0x, octal numbers with 0o and binary numbers with 0b. For example: 0×7f + 0b10010101.
Here are calculations that involve units.

- [ 2 meters + 5 feet ]

Convert from one set of units to another by using the notation, \(x\) units in \(y\) units.

- [ three quarters of a cup in teaspoons ]
- [ 98.6 degrees Fahrenheit in degrees Celsius ]
- [ 130 lbs in kg ]
- [ 65 mph in kph ] or [ 65 mph in km/h ]

Convert from one numbering system to another.

- [ 1500 in hex ] or [ 1500 in hexadecimal ]
- [ 64 in binary ]
MAKING A SEARCH ENGINE
What does it take to build a search engine?

- Decide what to index
- Collect it
- Index it (efficiently)
- Keep the index up to date
- Provide user-friendly query facilities
What else?

- Understand the structure of the web for efficient crawling
- Understand user information needs
- Preprocess text and other unstructured data
- Cluster data
- Classify data
- Evaluate performance
  - Next chapter
Thanks