Challenge: “The need behind the query”

- Semantic analysis
  - Query language determination
    - Auto filtering
    - Different ranking (if query in Japanese do not return English)
  - Hard & soft (partial) matches
    - Personalities (triggered on names)
    - Cities (travel info, maps)
    - Medical info (triggered on names and/or results)
    - Stock quotes, news (triggered on stock symbol)
    - Company info
    - Etc.
  - Natural Language reformulation
  - Integration of Search and Text Analysis
Answering “the need behind the query”: Context

• Context determination
  – spatial (user location/target location)
  – query stream (previous queries)
  – personal (user profile)
  – explicit (user choice of a vertical search, )
  – implicit (use Google from France, use google.fr)

• Context use
  – Result restriction
    • Kill inappropriate results
  – Ranking modulation
    • Use a “rough” generic ranking, but personalize later

The spatial context: Geo-search

• Two aspects
  – Geo-coding -- encode geographic coordinates to make search effective
  – Geo-parsing -- the process of identifying geographic context.

• Geo-coding
  – Geometrical hierarchy (squares)
  – Natural hierarchy (country, state, county, city, zip-codes, etc)

• Geo-parsing
  – Pages (infer from phone nos, zip, etc). About 10% can be parsed.
  – Queries (use dictionary of place names)
  – Users
    • Explicit (tell me your location -- used by NL, registration, from ISP)
    • From IP data
  – Mobile phones
    • Many sources of highly accurate location
Answering “the need behind the query”: Context

• Context determination
  – spatial (user location/target location)
  – query stream (previous queries)
  – personal (user profile)
  – explicit (user choice of a vertical search,)
  – implicit (use Google from France, use google.fr)

• Context use
  – Result restriction
    • Kill inappropriate results
  – Ranking modulation
    • Use a “rough” generic ranking, but personalize later

Context transfer
No transfer

Context transfer
1. An analysis of data characteristics that affect naive Bayes performance (PDF)

   An analysis of data characteristics that affect naive Bayes performance... ally a better indicator of naive Bayes performance than the...
   www.research.ibm.com/PM/rcml01.pdf - 357k - View as HTML - More from this site - Save - Block

2. Improving the Performance of Naive Bayes for Text Classification (PDF)

   Improving the Performance of Naive Bayes for Text Classification. Yirong Shen and Jing Jiang. CS224N Spring 2003 Abstract. We seek to improve the performance of the naive Bayes classifier
   www-nlp.stanford.edu/courses/cs224n/2003sp/yirong99/report.pdf - 135k - View as HTML - More from this site - Save - Block
Answering “the need behind the query”: Context

- Context determination
  - spatial (user location/target location)
  - query stream (previous queries)
  - personal (user profile)
  - explicit (user choice of a vertical search, e.g., Amazon or eBay)
  - implicit (use Google from France, use google.fr)

- Context use
  - Result restriction
    - Kill inappropriate results
  - Ranking modulation
    - Use a “rough” generic ranking, but personalize later

Result Restriction

- Geographic restrictions
  - Holocaust denial in Germany
  - Imagery that may be illegal in some jurisdictions, accepted in others

- Age restrictions
  - COPPA
Web Crawler

- Finds and downloads web pages automatically
  - provides the collection for searching
- Web is huge and constantly growing
- Web is not under the control of search engine providers
- Web pages are constantly changing
- Crawlers also used for other types of data

Retrieving Web Pages

- Web crawler client program connects to a domain name system (DNS) server
- DNS server translates the hostname into an internet protocol (IP) address
- Crawler then attempts to connect to server host using specific port
- After connection, crawler sends an HTTP request to the web server to request a page
  - usually a GET request
Crawling the Web

Web Crawler

- Starts with a set of *seeds*, which are a set of URLs given to it as parameters
- Seeds are added to a URL request queue
- Crawler starts fetching pages from the request queue
- Downloaded pages are parsed to find link tags that might contain other useful URLs to fetch
- New URLs added to the crawler’s request queue, or *frontier*
Processing steps in crawling

- Pick a URL from the frontier
- Fetch the document at the URL
- Parse the URL
  - Extract links from it to other docs (URLs)
- Check if URL has content already seen
  - If not, add to indexes
- For each extracted URL
  - Ensure it passes certain URL filter tests
  - Check if it is already in the frontier (duplicate URL elimination)

Basic crawl architecture

Which one?

[Diagram showing the flow from WWW to URL Frontier through DNS, Fetch, Parse, Content seen?, URL filter, robots filters, and URL set with Dup URL elim.]
What any crawler **must** do

- Be **Robust**: Be immune to spider traps and other malicious behavior from web servers
- Be **Polite**: Respect implicit and explicit politeness considerations

What any crawler **should** do

- Be capable of **distributed** operation: designed to run on multiple distributed machines
- Be **scalable**: designed to increase the crawl rate by adding more machines
- **Performance/efficiency**: permit full use of available processing and network resources