From simple search to search intelligence: the evolution of search engines

Jian-Yun Nie
University of Montreal
Canada
http://www.iro.umontreal.ca/~nie

Goals of this tutorial

- Importance of IR:
  - IR (search engine) is used for different tasks in our everyday life.
  - It is also used as a basic tool for other tasks (datamining, data analytics, QA, etc.)

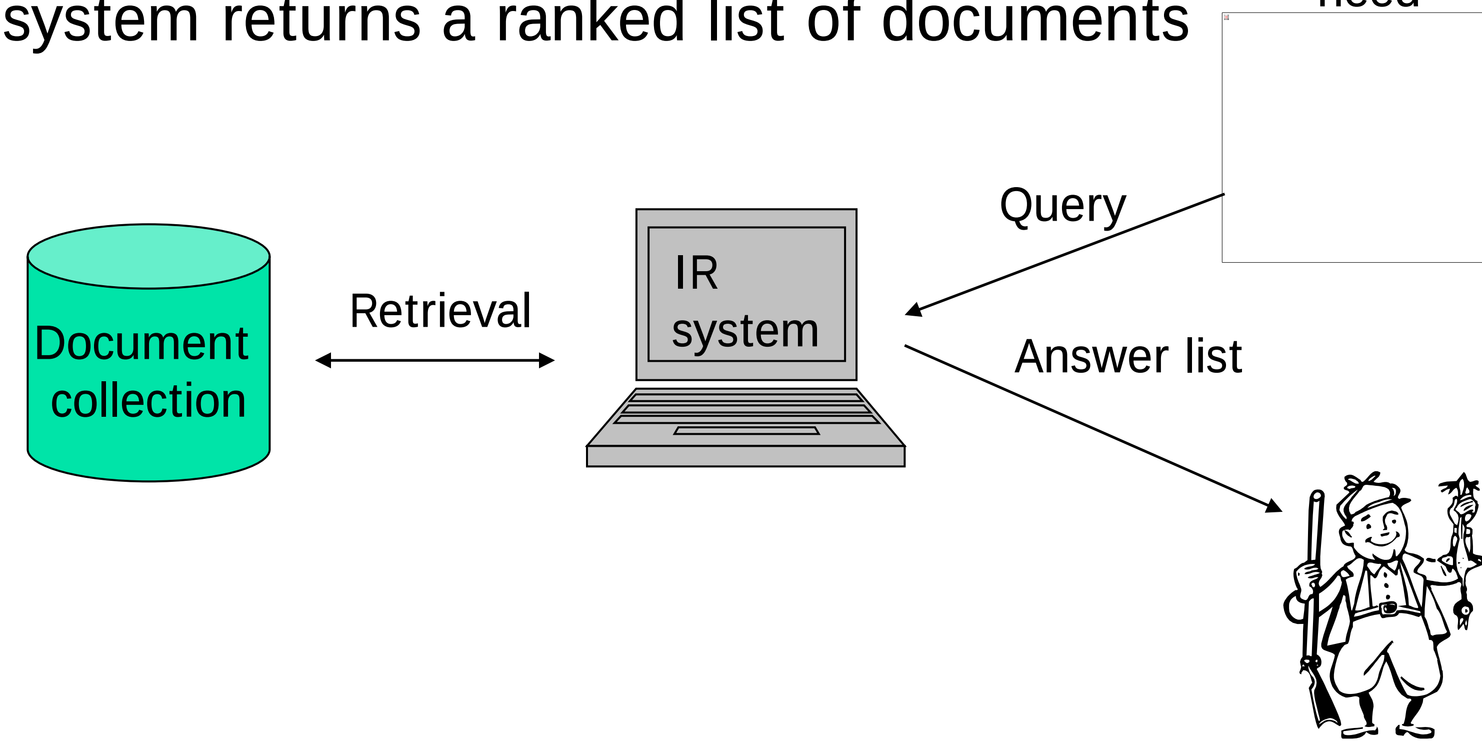
- This tutorial:
  - Understand how IR works
  - Common methods used
  - Recent evolution to do more than search

Outline

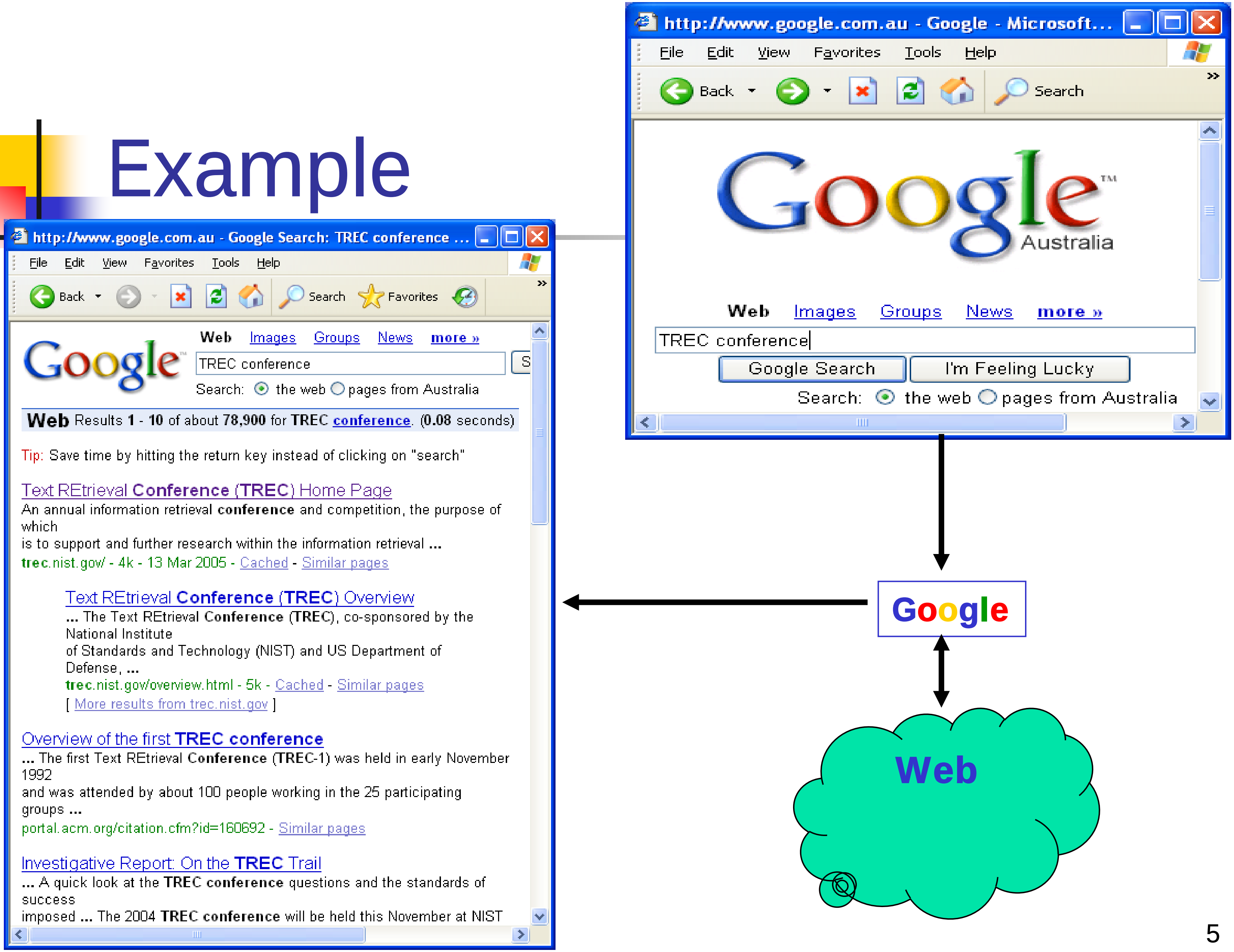
- IR problem and basic processing
- Traditional models
- More recent models
  - Links between documents and queries
  - User feedback
- Understanding the user (user's intent)
- Remaining challenges

The problem of IR

1. A user is in need of some information
2. He formulates a query to an IR system
3. IR system evaluates the relevance of documents
4. IR system returns a ranked list of documents



Example



http://www.google.com.au - Google - Microsoft...
File Edit View Favorites Tools Help
Back Search
Google Australia
Web Images Groups News more »
TREC conference
Google Search I'm Feeling Lucky
Search: the web pages from Australia
http://www.google.com.au - Google Search: TREC conference ...
File Edit View Favorites Tools Help
Back Search Favorites
Google
Web Images Groups News more »
TREC conference
Search: the web pages from Australia
Web Results 1 - 10 of about 78,900 for TREC conference. (0.08 seconds)
Tip: Save time by hitting the return key instead of clicking on "search"
Text REtrieval Conference (TREC) Home Page
An annual information retrieval conference and competition, the purpose of which
is to support and further research within the information retrieval ...
trec.nist.gov/ - 4k - 13 Mar 2005 - Cached - Similar pages
Text REtrieval Conference (TREC) Overview
... The Text REtrieval Conference (TREC), co-sponsored by the National Institute
of Standards and Technology (NIST) and US Department of Defense, ...
trec.nist.gov/overview.html - 5k - Cached - Similar pages
[More results from trec.nist.gov]
Overview of the first TREC conference
... The first Text REtrieval Conference (TREC-1) was held in early November 1992
and was attended by about 100 people working in the 25 participating groups ...
portal.acm.org/citation.cfm?id=160692 - Similar pages
Investigative Report: On the TREC Trail
... A quick look at the TREC conference questions and the standards of success
imposed ... The 2004 TREC conference will be held this November at NIST
Google
Web

Indexing-based IR

Document → **indexing** → Representation (keywords)

Query → **indexing** (Query analysis) → Representation (keywords)

Representation (keywords) ↔ Representation (keywords)

Retrieval

(Document ranking)

Basic problems in IR

- Document and query indexing
  - How to best represent their contents?
- Query evaluation (or retrieval process)
  - To what extent does a document correspond to a query?
  - A ranking/scoring function

Document indexing

- Goal
  - Identify the important content and create an internal representation for it
- What indexing units to use?
  - Words or Word stems (bag-of-words)
  - Phrases
  - Concepts
- How to weight?

Document indexing

- General process:
  - Input text
  - Processing of text format and structure
    - Word
    - index some fields and discard others
  - For each word form
    - Stopword?
    - Stemming
  - Term weighting

Stopwords / Stoplist

- Stopwords = words that do not bear useful information for IR

  e.g. of, in, about, with, I, although, ...
- Stoplist: contain stopwords, excluded from index
  - Prepositions
  - Articles
  - Pronouns
  - Some adverbs and adjectives
  - Some frequent words (e.g. document)

- The removal of stopwords usually improves slightly IR effectiveness, or not
- A few “standard” stoplists are commonly used.

Stemming

- Why?
  - Different word forms may bear similar meaning e.g. search, searching
- Stemming:
  - Removing some endings of word

computer
compute
computes
computing
computed
computation

comput

Porter algorithm

(Porter, M.F., 1980, An algorithm for suffix stripping, *Program*, **14**(3) :130-137)

- Step 1: plurals and past participles
  - SSES -> SS — caresses -> caress
  - (*v*) ING -> — motoring -> motor
- Step 2: adj->n, n->v, n->adj, ...
  - (m>0) OUSNESS -> OUS — callousness -> callous
  - (m>0) ATIONAL -> ATE — relational -> relate
- Step 3:
  - (m>0) ICATE -> IC — triplicate -> triplic
- Step 4:
  - (m>1) AL -> — revival -> reviv
  - (m>1) ANCE -> — allowance -> allow
- Step 5:
  - (m>1) E -> — probate -> probat
  - (m > 1 and *d and *L) -> single letter — controll -> control

Lemmatization – an alternative

- transform to standard form according to syntactic category.

  E.g. verb + *ing* $\rightarrow$ verb

  noun + *s* $\rightarrow$ noun

  - Need POS tagging
  - More accurate than stemming, but needs more resources

- crucial to choose stemming/lemmatization rules

  noise v.s. recognition rate

- compromise between precision and recall

light/no stemming ⟷ aggressive stemming

-recall +precision ⟷ +recall -precision

Traditional tf*idf weighting schema

- tf = term frequency
  - frequency of a term/keyword in a document

  The higher the tf, the higher the importance (weight) for the doc.
- df = document frequency
  - no. of documents containing the term
  - distribution of the term
- idf = inverse document frequency
  - the unevenness of term distribution in the corpus
  - the specificity of term to a document

  The more the term is distributed evenly, the less it is specific to a document

weight(t,D) = tf(t,D) * idf(t)

Some common *tf*idf* schemes

- tf(t, D)=freq(t,D)
- tf(t, D)=log[freq(t,D)]
- tf(t, D)=log[freq(t,D)]+1
- tf(t, D)=freq(t,d)/Max[f(t,d)]

idf(t) = log(N/n)

n = #docs containing t

N = #docs in corpus

weight(t,D) = tf(t,D) * idf(t)

- Normalization: Cosine normalization, /max, ...

Result of indexing

- Each document is represented by a set of weighted keywords (terms):

  $D_1 \rightarrow \{(t_1, w_1), (t_2,w_2), \ldots\}$

  e.g. $D_1 \rightarrow \{(\text{comput}, 0.2), (\text{architect}, 0.3), \ldots\}$

  $D_2 \rightarrow \{(\text{comput}, 0.1), (\text{network}, 0.5), \ldots\}$

- Inverted file:

  $\text{comput} \rightarrow \{(D_1,0.2), (D_2,0.1), \ldots\}$

  Inverted file is used during retrieval for higher efficiency.

Indexing-based IR

Document

indexing

Representation
(keywords)

Query

indexing

(Query analysis)

Representation
(keywords)

Retrieval

(Document ranking)

Retrieval

- The problems underlying retrieval
  - Retrieval model
    - What is the formal representation by the extracted and weighted terms?
    - How to match a query representation with a document representation to estimate a score?
  - Many models have been proposed in IR

Traditional Models

Boolean model

- Document = Logical conjunction of keywords
- Query = Boolean expression of keywords
- $R(D, Q) = D \rightarrow Q$

e.g. $D = t_1 \wedge t_2 \wedge \ldots \wedge t_n$
$Q = (t_1 \wedge t_2) \vee (t_3 \wedge \neg t_4)$
$D \rightarrow Q$, thus $R(D, Q) = 1$.

Problems:

- R is either 1 or 0 – No ranking
- Result in many documents or few documents

- Often used as a first filtering of result candidates in search engines

Vector space model

- Vector space = all the keywords encountered

  $<t_1, \quad t_2, \quad t_3, \ldots, t_n>$
- Document

  $D = \quad < a_1, a_2, a_3, \ldots, a_n>$

  a_i = weight of t_i in D
- Query

  $Q = \quad < b_1, b_2, b_3, \ldots, b_n>$

  b_i = weight of t_i in Q

- R(D,Q) = Sim(D,Q)

Some formulas for Sim

Dot product

$$Sim(D,Q) = D \bullet Q = \sum_i (a_i * b_i)$$

Cosine

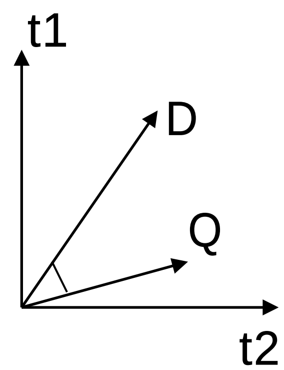
$$Sim(D,Q) = \frac{\sum_i (a_i * b_i)}{\sqrt{\sum_i a_i^2 * \sum_i b_i^2}}$$

Dice

$$Sim(D,Q) = \frac{2\sum_i (a_i * b_i)}{\sum_i a_i^2 + \sum_i b_i^2}$$

Jaccard

$$Sim(D,Q) = \frac{\sum_i (a_i * b_i)}{\sum_i a_i^2 + \sum_i b_i^2 - \sum_i (a_i * b_i)}$$



Probabilistic model

- Given D and Q, estimate P(R|D,Q) and P(NR|D,Q)

$$P(R \mid Q, D) = \frac{P(D \mid R, Q) * P(R \mid Q)}{P(D \mid Q)}$$

  - P(D|Q), P(R|Q) assumed constant

So, $P(R|Q,D) \propto P(D|R)$

Probabilistic model

- Binary independent model

$$D = \{t_1=x_1,\ t_2=x_2,\ \ldots\}$$

$$x_i = \begin{cases} 1 & present \\ 0 & absent \end{cases}$$

$$P(D \mid R, Q) = \prod_{(t_i = x_i) \in D} P(t_i = x_i \mid R_Q)$$

$$= \prod_{t_i} p_i^{x_i} (1 - p_i)^{(1 - x_i)}$$

$$p = P(t_i = 1 \mid R_Q)$$

$$q = P(t_i = 0 \mid R_Q)$$

$$P(D \mid NR, Q) = \prod_{t_i} P(t_i = 1 \mid NR_Q)^{x_i} P(t_i = 0 \mid NR_Q)^{(1 - x_i)} = \prod_{t_i} q_i^{x_i} (1 - q_i)^{(1 - x_i)}$$

Prob. model (cont'd)

For document ranking

$$Odd(D) = \log \frac{P(D \mid R, Q)}{P(D \mid NR, Q)} = \log \frac{\prod_{t_i} p_i^{x_i} (1 - p_i)^{(1-x_i)}}{\prod_{t_i} q_i^{x_i} (1 - q_i)^{(1-x_i)}}$$

$$= \sum_{t_i} x_i \log \frac{p_i (1 - q_i)}{q_i (1 - p_i)} + \sum_{t_i} \log \frac{1 - p_i}{1 - q_i}$$

Constant for any document

$$\propto \sum_{t_i} x_i \log \frac{p_i (1 - q_i)}{q_i (1 - p_i)}$$

Prob. model (cont'd)

- How to estimate p_i and q_i?
- A set of N relevant and irrelevant samples:

$$p_i = \frac{r_i}{R_i} \qquad q_i = \frac{n_i - r_i}{N - R_i}$$

r_i Rel. doc. with t_i	n_i-r_i Irrel.doc. with t_i	n_i Doc. with t_i
R_i-r_i Rel. doc. without t_i	N-R_i–n+r_i Irrel.doc. without t_i	N-n_i Doc. without ti
R_i Rel. doc	N-R_i Irrel.doc.	N Samples

Prob. model (cont'd)

$$Odd(D) = \sum_{t_i} x_i \log \frac{p_i(1-q_i)}{q_i(1-p_i)} = \sum_{t_i} x_i \log \frac{r_i(N - R_i - n_i + r_i)}{(R_i - r_i)(n_i - r_i)}$$

- Smoothing (Robertson-Sparck-Jones formula)

$$Odd(D) = \sum_{t_i} x_i \log \frac{(r_i + 0.5)(N - R_i - n_i + r_i + 0.5)}{(R_i - r_i + 0.5)(n_i - r_i + 0.5)} = \sum_{t_i \in D} x_i w_i$$

- When no sample is available:
  $p_i=0.5$,
  $q_i=(n_i+0.5)/(N+0.5)\approx n_i/N$

$$w_i = \log \frac{0.5(1-\frac{n_i}{N})}{\frac{n_i}{N}(1-0.5)} = \log \frac{N-n_i}{n_i} \quad (\sim\text{idf})$$

- Use relevance feedback to get more samples for more precise estimation

BM25 – one of the best performing models

$$Score(D,Q) = \sum_{t \in Q} w \frac{(k_1+1)tf}{K+tf} \frac{(k_3+1)qtf}{k_3+qtf} + k_2 |Q| \frac{avdl - dl}{avdl + dl}$$

$$K = k_1((1-b) + b\frac{dl}{avdl - dl})$$

TF factors

Doc. length normalization

- k1, k2, k3, d: parameters
- qtf: query term frequency
- dl: document length
- avdl: average document length

Statistical Language models for IR

Basics: Prob. of a sequence of words

$$P(s) = P(w_1, w_2, \ldots w_n)$$
$$= P(w_1)P(w_2 \mid w_1)\ldots P(w_n \mid w_{1,n-1})$$
$$= \prod_{i=1}^{n} P(w_i \mid h_i)$$

Elements to be estimated:

$$P(w_i \mid h_i) = \frac{P(h_i w_i)}{P(h_i)}$$

- If *hi* is too long, one cannot observe (*hi, wi*) in the training corpus, and (*hi, wi*) is hard generalize

- Solution: limit the length of *hi*

N-grams

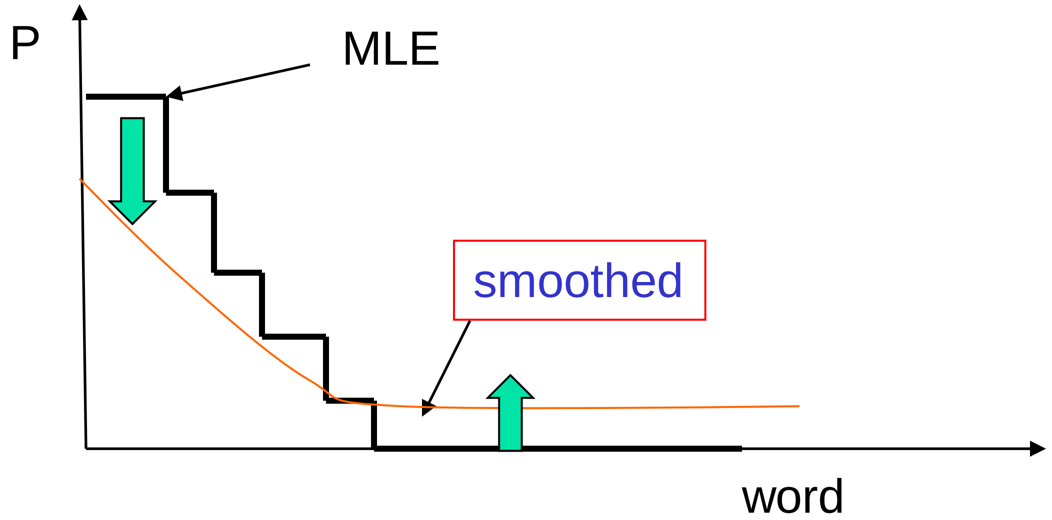
- Limit h_i to n-1 preceding words

  Most used cases

  - Uni-gram: $$P(s) = \prod_{i=1}^{n} P(w_i)$$
  - Bi-gram: $$P(s) = \prod_{i=1}^{n} P(w_i \mid w_{i-1})$$
  - Tri-gram: $$P(s) = \prod_{i=1}^{n} P(w_i \mid w_{i-2} w_{i-1})$$

Smoothing

- Problem of data sparsity:
  - An n-gram may not be observed from a training data
  - This does not mean that the n-gram is impossible in the language
- Smoothing = assign a small probability to unobserved words or n-grams



Smoothing methods

n-gram: α

- Change the freq. of occurrences
  - Laplace smoothing (add-one):

$$P_{add\_one}(\alpha \mid C) = \frac{freq(\alpha)+1}{\sum_{\alpha_i \in V} (freq(\alpha_i)+1)}$$

  - Does not work well
    - A large part of probability mass is assigned due to smoothing

Smoothing (cont'd)

- Combine a model with a lower-order model
  - Interpolation (Jelinek-Mercer)

$$P_{JM}(w_i \mid w_{i-1}) = \lambda_{w_{i-1}} P_{ML}(w_i \mid w_{i-1}) + (1-\lambda_{w_{i-1}}) P_{JM}(w_i)$$

- In IR, combine doc. with corpus
  - Interpolation

$$P(w_i \mid D) = \lambda P_{ML}(w_i \mid D) + (1-\lambda) P_{ML}(w_i \mid C)$$

  - Dirichlet

$$P_{Dir}(w_i \mid D) = \frac{tf(w_i, D) + \mu P_{ML}(w_i \mid C)}{|D| + \mu}$$

Using LM in IR: Query generation

- Rank document D according to its model's capacity to generate the query Q, i.e. $P(Q|D)$
- But we want to rank according to $P(D|Q)$?
- $P(D \mid Q) = P(Q \mid D) \; * \; P(D) \; / \; P(Q)$
  - $P(Q)$ is the same for all documents, so ignore
  - $P(D)$ [the prior] is often treated as the same for all D
    - But we could use criteria like authority, length, genre
  - So, $P(D|Q) \sim P(Q|D)$

Query generation

- Another explanation:
  - Before submitting a query, the user imagines some ideal document D_{ideal} he would like to find, and the words that would appear in the document
  - Q is formed using these words
  - So the user is generating Q from D_{ideal}
  - Submit D to the same generation process
    - If P(Q|D) is high, then D is close to D_{ideal}

Using LM in IR: Divergence between models

Question: Is the document likelihood increased when a query is submitted?

$$LR(D,Q) = \frac{P(D \mid Q)}{P(D)} = \frac{P(Q \mid D)}{P(Q)}$$

(Is the query likelihood increased when D is retrieved?)

- $P(Q|D)$ calculated with $P(Q|M_D)$
- $P(Q)$ estimated as $P(Q|M_C)$

$$Score(Q,D) = \log \frac{P(Q \mid M_D)}{P(Q \mid M_C)}$$

Divergence between M_D and M_Q

Assume Q follows a multinomial distribution :

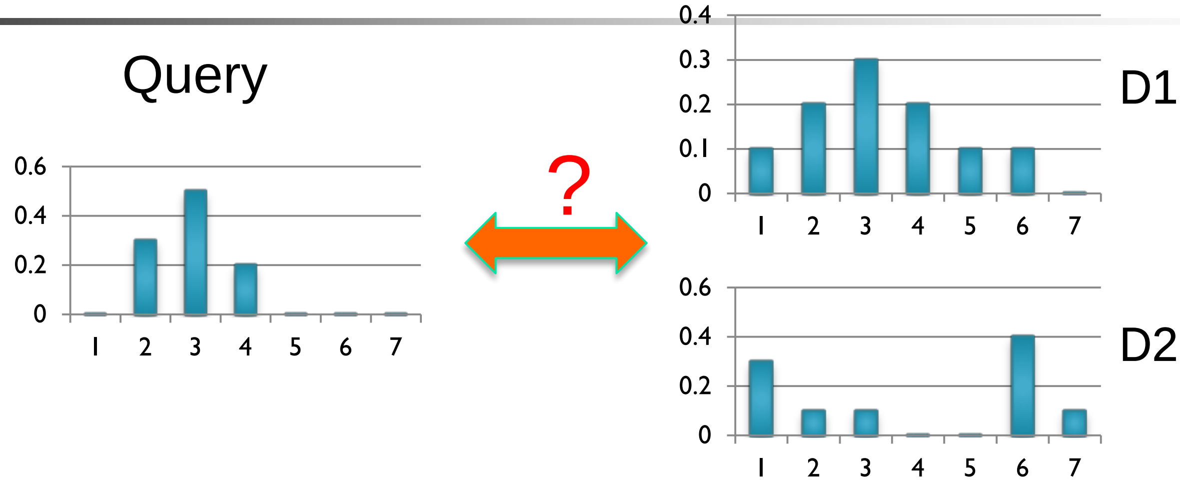
$$P(Q \mid M_D) = \frac{|Q|!}{\prod_{q_i \in Q} tf(q_i, Q)!} \prod_{q_i \in Q} P(q_i \mid M_D)^{tf(q_i, Q)}$$

$$P(Q \mid M_C) = \frac{|Q|!}{\prod_{q_i \in Q} tf(q_i, Q)!} \prod_{q_i \in Q} P(q_i \mid M_C)^{tf(q_i, Q)}$$

$$\begin{aligned}
Score(Q, D) &= \sum_{i=1}^{n} tf(q_i, Q) * \log \frac{P(q_i \mid M_D)}{P(q_i \mid M_C)} \\
&\propto \sum_{i=1}^{n} P(q_i \mid M_Q) * \log \frac{P(q_i \mid M_D)}{P(q_i \mid M_C)} \\
&= \sum_{i=1}^{n} P(q_i \mid M_Q) * \log \frac{P(q_i \mid M_D)}{P(q_i \mid M_Q)} - \sum_{i=1}^{n} P(q_i \mid M_Q) * \log \frac{P(q_i \mid M_C)}{P(q_i \mid M_Q)} \\
&= \sum_{i=1}^{n} P(q_i \mid M_Q) * \log P(q_i \mid M_D) + \text{Constant} \\
&\propto -KL(M_Q, M_D)
\end{aligned}$$

Negative Kullback-Leibler divergence: larger KL-divergence = lower rank

Another view of model divergence



$$Score(Q,D) = \sum_{i=1}^{n} P(q_i \mid \theta_Q) * \log P(q_i \mid \theta_D)$$

Query model

Document model

Comparaison: LM v.s. tf*idf

$$P(Q|D)=\prod_i P(q_i|D)$$

$$=\prod_{q_i\in Q\cap D}(\lambda\frac{tf(q_i,D)}{|D|}+(1-\lambda)\frac{tf(q_i,C)}{|C|})\prod_{q_j\in Q-D}((1-\lambda)\frac{tf(q_j,C)}{|C|})$$

$$=\prod_{q_i\in Q\cap D}(\lambda\frac{tf(q_i,D)}{|D|}+(1-\lambda)\frac{tf(q_i,C)}{|C|})\prod_{q_j\in Q}((1-\lambda)\frac{tf(q_j,C)}{|C|})/\prod_{q_j\in Q\cap D}((1-\lambda)\frac{tf(q_j,C)}{|C|})$$

$$=const\prod_{q_i\in Q\cap D}(\frac{\lambda\frac{tf(q_i,D)}{|D|}+(1-\lambda)\frac{tf(q_i,C)}{|C|}}{(1-\lambda)\frac{tf(q_i,C)}{|C|}})=const\prod_{q_i\in Q\cap D}(\frac{\lambda}{1-\lambda}\frac{\frac{tf(q_i,D)}{|D|}}{\frac{tf(q_i,C)}{|C|}}+1)$$

idf

- Log P(Q|D) ~ VSM with tf*idf and document length normalization

- Smoothing ~ idf + length normalization

Common extensions

Some common techniques to improve IR effectiveness

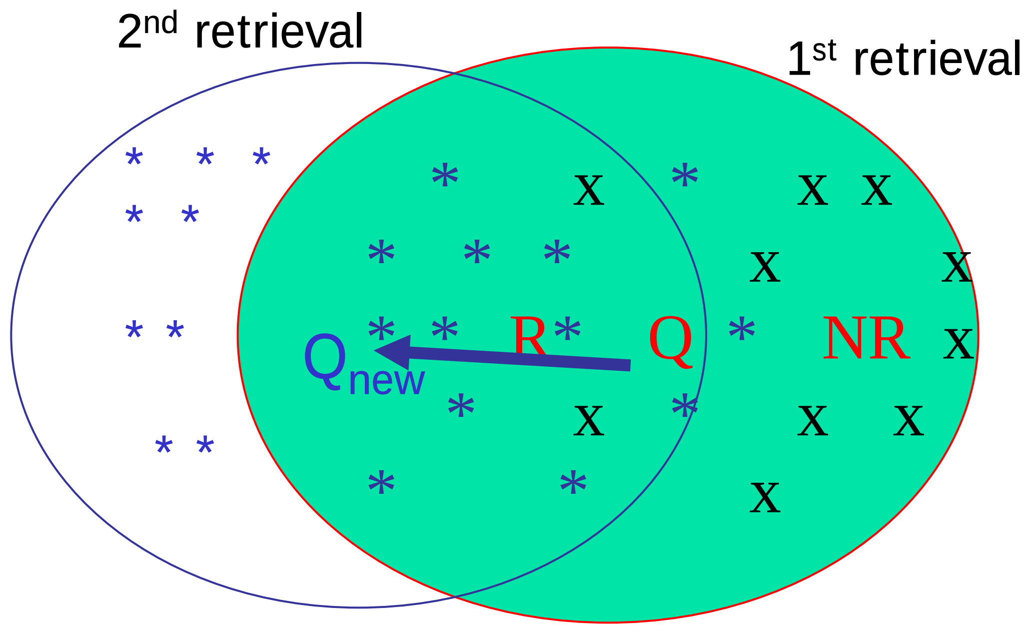
- Interaction with user (relevance feedback)
  - Keywords only cover part of the contents
  - User can help determining relevant/irrelevant document
- The use of relevance feedback
  - To improve query expression:

$$Q_{new} = \alpha * Q_{old} + \beta * Rel\_d - \gamma * Nrel\_d$$

where Rel_d = centroid of relevant documents

NRel_d = centroid of non-relevant documents

Effect of RF



Modified relevance feedback

- Users usually do not cooperate (e.g. AltaVista in early years)
- Pseudo-relevance feedback (Blind RF)
  - Using the top-ranked documents as if they are relevant (e.g. 20 documents)
    - Select m terms from n top-ranked documents
  - One can usually obtain about 5-10% improvement in MAP in TREC experiments

Query expansion

- A query contains part of the important words
- Add new (related) terms into the query
  - Manually constructed knowledge base/thesaurus (e.g. Wordnet)
    - Q = information retrieval
    - Q' = (information + data + knowledge + ...)
(retrieval + search + seeking + ...)
  - Co-occurrence analysis:
    - two terms that often co-occur are related (Mutual information)
    - Two terms that co-occur with the same words are related (e.g. T-shirt and coat with wear, ...)

Global vs. local context analysis [Xu and Croft]

- Global analysis: use the whole document collection to calculate term relationships
- Local analysis: use the query to retrieve a subset of documents, then calculate term relationships
  - Combine pseudo-relevance feedback and term co-occurrences
  - More effective than global analysis

Query Expansion Approach (1)

- Wordnet [Voorheers 1994]:
  - Using synonyms, hypernyms and hyponyms to expand query
  - Problems:
    - Coverage is low, only contains linguistically motivated relationships
    - Ambiguity: e.g. "*computer: machine* or *human expert*"
    - Lack of strength measure for relationships: weighting of expanded terms
  - Can not improve retrieval effectiveness

- Co-occurrence [Qiu 1993]:
  - Two words that often co-occur are related
  - Capable of extracting: e.g. "*Java → programming*"
  - Some improvements
  - Problems:
    - Introduce noise: frequent co-occurring terms are not necessarily related
    - No context information for term relationships: possible to expand "*Java travel*" by "*programming*"

Query Expansion Approach (2)

- Pseudo-relevant feedback [Zhai 2001]:
  - Retrieve some documents with query
  - Top *n* feedback documents are assumed to be relevant
  - Extract expansion terms from feedback documents
  - Most effective method so far
  - Problem: two retrieval processes => longer retrieval time

Inference in LM?

IR as an inference process

- Key: inference – infer query from document
  - D: Tsunami
  - Q: natural disaster
  - $D \rightarrow Q$?

Inference in traditional models?

1. Traditional bag-of-words approach:

   Matching words, no inference

2. Language model?

- P(Q|D) ~ P(D→Q)

- Smoothing:

$$P(t_i \mid D) = \lambda P_{ML}(t_i \mid D) + (1-\lambda) P_{ML}(t_i \mid C)$$

$$t_i \notin D : P_{ML}(t_i \mid D) = 0$$

change to $P(t_i \mid D) > 0$

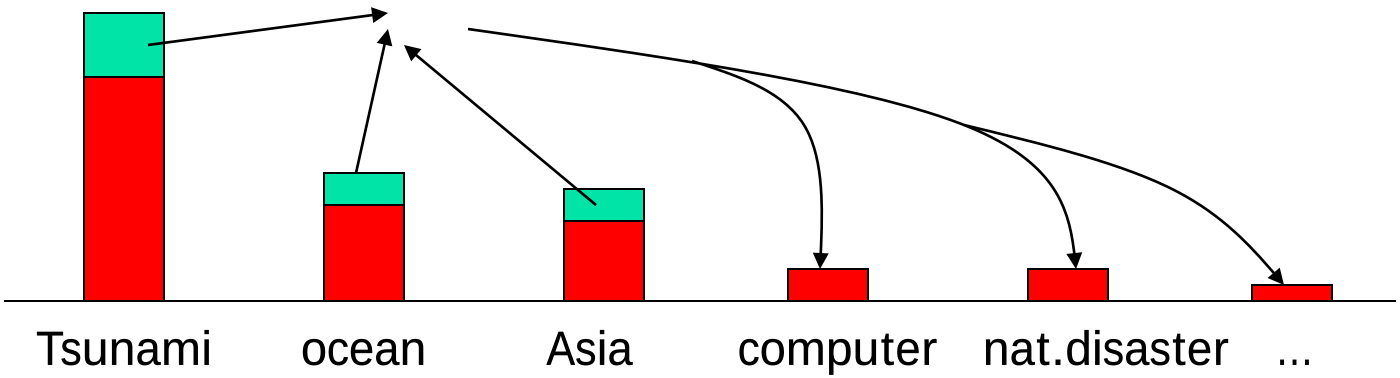
- E.g. D=Tsunami, P_{ML}(natural disaster|D)=0

  change to P(natural disaster|D)>0

- Inference by accident
  - P(computer|D)>0 ~ P(natural disaster|D)>0

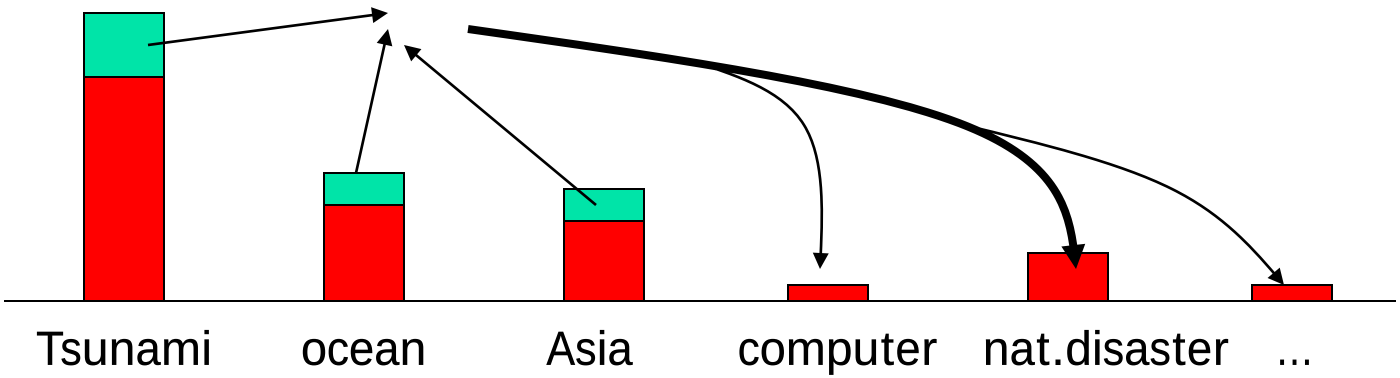
Effect of smoothing?

- Doc: Tsunami, ocean, Asia, ...



- Smoothing $\neq$ inference
- Redistribution uniformly/according to collection (also to unrelated terms)

Expected effect



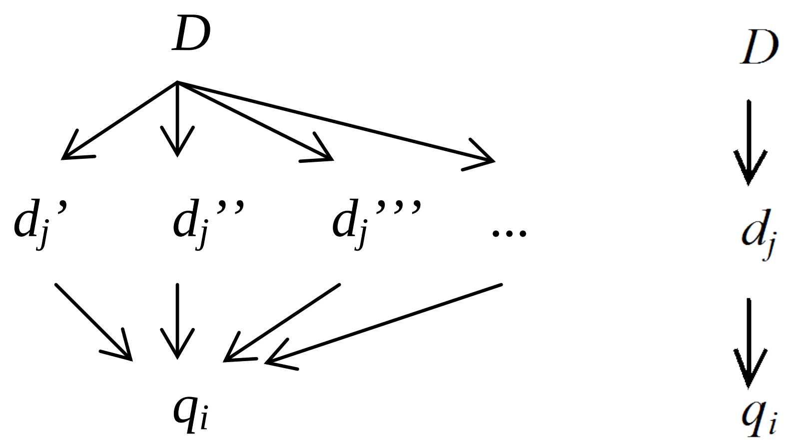
- Using Tsunami → natural disaster
- Knowledge-based smoothing

Inference – incorporating some knowledge in document model

- Translation model [Berger and Lafferty, 1999]

$$P(q_i \mid D) = \sum_{d_j} P(q_i \mid d_j) P(d_j \mid D)$$

$$P(Q \mid D) = \prod_{q_i} \sum_{d_j} P(q_i \mid d_j) P(d_j \mid D)$$



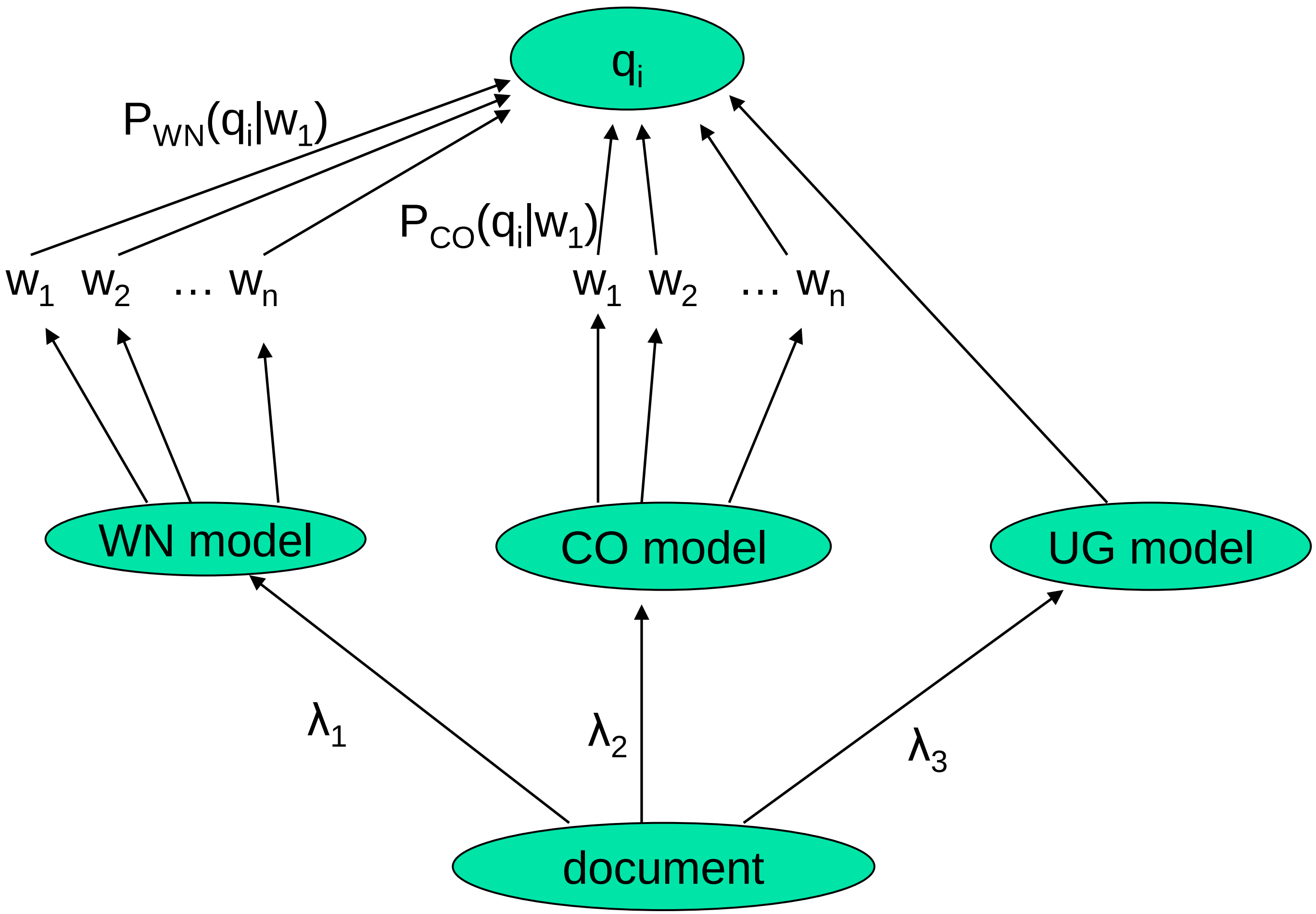
$$(D \rightarrow d_j) \wedge (d_j \rightarrow q_i) \mid\!- (D \rightarrow q_i)$$

Using multiple knowledge sources (Cao et al. 05)

- Different ways to satisfy a query (term)
  - Directly though unigram model
  - Indirectly (by inference) through Wordnet relations
  - Indirectly trough Co-occurrence relations
  - ...
- $D \rightarrow t_i$ if $D \rightarrow_{UG} t_i$ or $D \rightarrow_{WN} t_i$ or $D \rightarrow_{CO} t_i$

$$P(t_i \mid D) = \lambda_1 P_{UG}(t_i \mid D) + \lambda_2 \sum_j P_{WN}(t_i \mid t_j) P(t_j \mid D) + \lambda_3 \sum_j P_{CO}(t_i \mid t_j) P(t_j \mid D)$$

Inference using different types of knowledge (Cao et al. 05)



Incorporating knowledge in Query expansion

- KL-div: ~ $$\sum_{t_i \in Q} \underbrace{P(t_i \mid Q)}_{\text{Query model}} \log \underbrace{P(t_i \mid D)}_{\text{Smoothed doc. model}}$$
- With no query expansion, equivalent to generative model

$$-KL(Q \,\|\, D) \sim \sum_{t_i \in Q} P(t_i \mid Q) \log P(t_i \mid D)$$

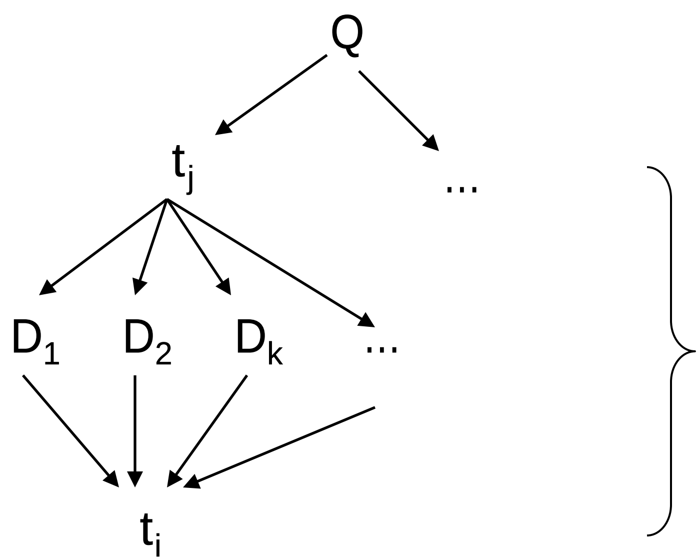
$$\sim \sum_{t_i \in Q} tf(t_i, Q) \log P(t_i \mid D) = \sum_{t_i \in Q} \log P(t_i \mid D)^{tf(t_i,Q)}$$

$$\sim P(Q \mid D)$$

Extension for inference: Query (relevance) model [Lavrenko & Croft 2001]

- Using pseudo feedback documents

$$P(t_i \mid Q) = \sum_{j,k} P(t_i \mid D_k) P(D_k \mid t_j) P_{ML}(t_j \mid Q)$$



$t_j \rightarrow t_i$, $P(t_i|t_j)$

- Some inference (~co-occurrence) through pseudo-feedback documents

Expanding query model

$$P(q_i \mid Q) = \lambda P_{ML}(q_i \mid Q) + (1-\lambda) P_R(q_i \mid Q)$$

$P_{ML}(t_j \mid Q)$: Max.Likelihood unigram model (not smoothed)

$P_R(t_i \mid Q)$: Relational model

$$Score(Q,D) = \sum_{q_i \in V} P(q_i \mid Q) \times \log P(q_i \mid D)$$

$$= \sum_{q_i \in V} [\lambda P_{ML}(q_i \mid Q) + (1-\lambda) P_R(q_i \mid Q)] \times \log P(q_i \mid D)$$

$$= \underbrace{\lambda \sum_{q_i \in Q} P_{ML}(q_i \mid Q) \times \log P(q_i \mid D)}_{\text{Classical LM}} + (1-\lambda) \underbrace{\sum_{q_i \in V} P_R(q_i \mid Q) \times \log P(q_i \mid D)}_{\text{Relation model}}$$

How to estimate $P_R(t_i \mid Q)$?

- Using co-occurrence information
- Using an external knowledge base (e.g. Wordnet)
- Pseudo-rel. feedback
- Other term relationships
- ...

Using co-occurrence relation

- Use term co-occurrence relationship
  - Terms that often co-occur in the same windows are related
  - Window size: 10 words
- Unigram relationship ($w_j \rightarrow w_i$)

$$P_R(w_i \mid w_j) = \frac{c(w_i, w_j)}{\sum_{w_l} c(w_l, w_j)}$$

- Query expansion

$$P_R(w_i \mid Q) = \sum_{q_j \in V} P_R(w_i, q_j \mid Q) = \sum_{q_j \in Q} P_R(w_i \mid q_j) \times P_{MLE}(q_j \mid Q)$$

Problems in co-occurrence relations

- Ambiguity
- Term relationship between two single words

  e.g. "*Java → programming*"
- No information to determine the appropriate context

  e.g. "*Java travel*" by "*programming*"
- Solution: add some context information into term relationship

Context-dependent expansion
(Bai et al. 06)

- Use *$(t_1, t_2, t_3, \ldots) \rightarrow t$* instead of *$t_1 \rightarrow t$*
  - e.g. "*(Java, computer, language) → programming*"

- Problem:
  - Complexity with many words in condition part
  - Difficult to obtain reliable relations

- A solution:
  - Limit condition part to 2 words
    e.g. "*(Java, computer) → programming*"
    "*(Java, travel) → island*"
  - One word specifies the context to the other

Context-dependent co-occurrences

- $w_i w_j \rightarrow w_k$

$$P_R(w_i \mid w_j w_k) = \frac{c(w_i, w_j, w_k)}{\sum_{w_l} c(w_l, w_j, w_k)}$$

- Bi-term relation model

$$P_R(w_i \mid Q) = \sum_{q_j, q_k \in V} P_R(w_i \mid q_j q_k) \times P(q_j q_k \mid Q) \approx \sum_{q_j, q_k \in Q} P_R(w_i \mid q_j q_k) \times P(q_j q_k \mid Q)$$

$$P(q_j q_k \mid Q) : \text{uniform}$$

Example

- Compare expansion terms by UQE and BQE:
  e.g. Query #55: "*Insider trading*"

  - Unigram relationships (UQE): *P(*|insider)* or *P(*|trading)*

stock:0.014177	market:0.0113156	US:0.0112784	year:0.010224
exchang:0.0101797	trade:0.00922486	report:0.00825644	price:0.00764028
dollar:0.00714267	1:0.00691906	govern:0.00669295	state:0.00659957
futur:0.00619518	million:0.00614666	dai:0.00605674	offici:0.00597034
peopl:0.0059315	york:0.00579298	issu:0.00571347	nation:0.00563911

  - Bi-term relationships (BQE): *P(*|insider, trading)*

secur:0.0161779	charg:0.0158751	stock:0.0137123	scandal:0.0128471
boeski:0.0125011	inform:0.011982	street:0.0113332	wall:0.0112034
case:0.0106411	year:0.00908383	million:0.00869452	investig:0.00826196
exchang:0.00804568	govern:0.00778614	sec:0.00778614	drexel:0.00756986
fraud:0.00718055	law:0.00631543	ivan:0.00609914	profit:0.00566658

Dependence Models for Information Retrieval

-- Dependency = the meaning of a word depends on another word

-- "computer architecture" ≠ computer + architecture

Dealing with dependencies in IR

- Quoted queries
  - "black Monday": OK
  - "computer architecture": ?
  - "淘宝手机": No
  - How can a user decide with little knowledge on IR and data?
- Phrases
  - Automatically detect phrases
    - Dictionary
    - Statistical analysis
  - Integration into an IR model

A typical integration method

- Detect phrases in the query and the documents
- Train a phrase model and a word model

$$Score(D,Q) = \alpha Score_{phrase}(D,Q) + (1-\alpha) Score_{word}(D,Q)$$

- Limited impact
  – [Fagan 88]: syntactic phrases have less impact than statistical phrases
  – More recent work tends to confirm

Why?

- Not all phrases are fixed expressions
  - “Black Monday” is
  - but not “desktop computer” and “HD movie”
    - desktop computer ≈ desktop
    - HD movie ≈ movie in HD ≈ HD…movie
- Many user queries are not grammatical
  - Bag of words
  - ≠ strict and unique expression of query intent
  - Assuming them to be fixed expressions hurts IR

How to tackle flexible dependencies?

- Consider n-grams, not phrases [Bai et al. WWW'08]
- Dependency model [Gao et al. 04]
- Term proximity [Tao and Zhai 07] [Zhao and Yun, 09]
- Consider adjacent terms or all terms in a query [Metzler and Croft 05]

Matching N-grams

- Given a long query *abcdefg*
- Extract n-grams: *abc, bcd, cde, def, efg*
- An extra score for a document according to its matches to the n-grams

$$Score(D,Q) = \alpha Score_{ngram}(D,Q) + (1-\alpha)Score_{baseline}(D,Q)$$

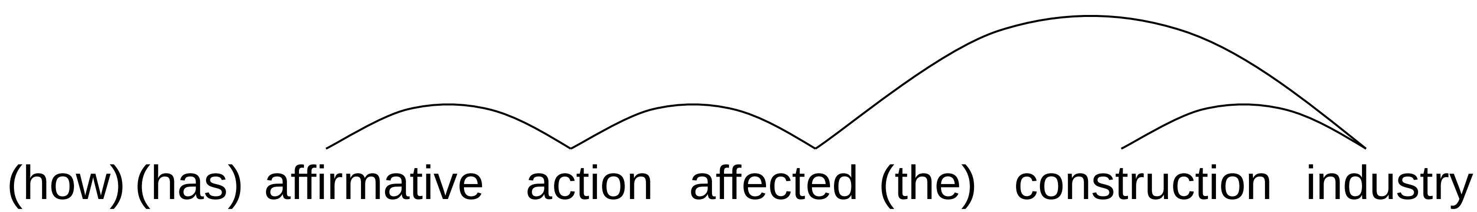
- Not all n-grams are meaningful and useful
- Limited impact

Dependence Model

- Dependence LM (Gao et al. 04)

  Capture more distant dependencies within a sentence

  - Syntactic analysis
  - Statistical analysis
    - Only retain the most probable dependencies in the query



Estimate the prob. of links (EM)

For a corpus C:

1. Initialization: link each pair of words with a window of 3 words
2. For each sentence in C:
   - Apply the link prob. to select the strongest links that cover the sentence
3. Re-estimate link prob.
4. Repeat 2 and 3

Calculation of P(Q|D)

1. Determine the links in Q (the required links)

$$L = \arg\max_L P(L \mid Q) = \arg\max_L \prod_{(i,j)\in L} P_C(R \mid q_i, q_j)$$

2. Calculate the likelihood of Q (words and links)

$$P(Q \mid D) = P(L \mid D) P(Q \mid L, D)$$

$$\left.P(L \mid D) = \prod_{l \in L} P(l \mid D) \qquad \right\} \textbf{links}$$

$$P(Q \mid L, D) = P(q_h \mid D) \prod_{(i,j)\in L} P(q_j \mid q_i, L, D) = \ldots$$

$$= \prod_{i=1 \ldots n} \underbrace{P(q_i \mid D)}_{\text{words}} \prod_{(i,j)\in L} \underbrace{\frac{P(q_i, q_j \mid L, D)}{P(q_i \mid D) P(q_j \mid D)}}_{\text{bi-terms}}$$

Requirement on words and bi-terms

Term proximity

- If two query term appear closely in document, then higher score
  - Calculate a span of query terms [Tao and Zhai]
    - Increase the score if span is small
  - Smooth the document model by *term centrality* [Zhao and Yun]

$$\hat{\theta}^{B}_{D,i} = \frac{c(w_i; D) + \lambda Prox_B(w_i) + \mu P(w_i|C)}{|D| + \sum_{j=1}^{|V|} \lambda Prox_B(w_j) + \mu}$$

$$P\_SumProx(q_i) = \sum_{q_j \in Q, q_j \neq q_i} f\left(Dis(q_i, q_j; D)\right)$$

$$f(x) = para^{-x}$$

Markov Random Field

- For a graph G, the joint probability:

$$P(X_1, \ldots, X_n) = \frac{1}{Z} \prod_{c \in C(G)} \psi_c(X_c)$$

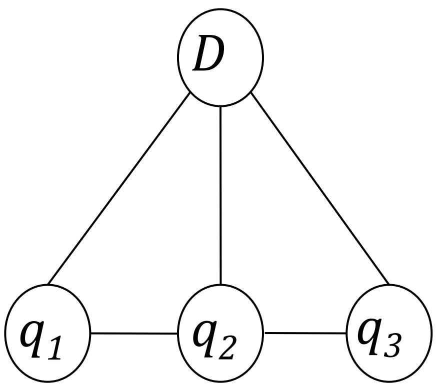
$X_1, \ldots, X_n$: random variables

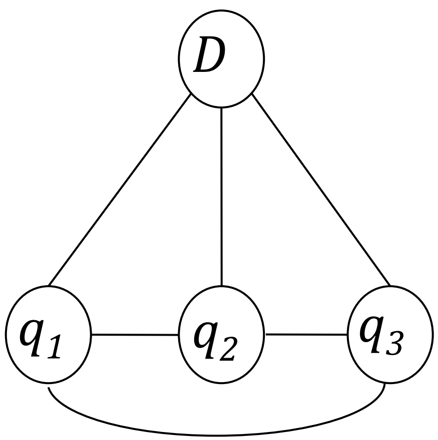
$\psi_c(X_c)$: potential function on a set of variables X_c

$C(G)$: set of cliques

Markov Random Field for IR
(Metzler and Croft, 2005)

- Two graphs for IR





Sequential dependence   Full dependence

$$Score(D,Q) = P_{\Lambda}(Q,D) = \frac{1}{Z} \prod_{c \in C(G)} \psi(c;\Lambda)$$

Markov Random Field for IR

$$Score(D,Q) = P_\Lambda(Q,D) = \frac{1}{Z}\prod_{c \in C(G)} \psi(c;\Lambda)$$

$$\stackrel{rank}{=} \sum_{c \in C(G)} \lambda_c f(c) = \sum_{c \in T} \lambda_T f_T(c) + \sum_{c \in O} \lambda_O f_O(c) + \sum_{c \in U} \lambda_U f_U(c)$$

$f_T(c) =$

- 3 components:
  - Term – T $\quad f_T(c) = \log P(q_i \mid D)$
  - Ordered term clique – O $\quad f_O(c) = \log P(\#1(q_i, \ldots, q_{i+k}) \mid D)$
  - Unordered term clique – U $\quad f_U(c) = \log P(\#uw(q_i, \ldots, q_{i+k}) \mid D)$
- λ_T, λ_O, λ_U : fixed parameters
  - Typical good values (0.8, 0.1, 0.1) or (0.85, 0.1, 0.05)

Weighted MRF-SD (Bendersky, Croft and Metzler, 10)

- Make the parameters λ_T, λ_O, λ_U dependent on the pair of terms

$$\lambda(q_i) = \sum_{j=1}^{k_{uni}} w_j^{uni} g_j^{uni}(q_i)$$

$$\lambda(q_i, q_{i+1}) = \sum^{k_{bi}} w_i^{bi} g_i^{bi}(q_i, q_{i+1})$$

$$P(D|Q) \overset{rank}{=} \sum_{j=1}^{k_{uni}} w_j^{uni} \sum_{q_i \in Q} g_j^{uni}(q_i) f_T(q, D) + \sum_{j=1}^{k_{bi}} w_j^{bi} \sum_{q_i q_{i+1} \in Q} g_j^{bi}(q_i, q_{i+1}) [f_O(q_i q_{i+1}, D) + f_U(q_i q_{i+1}, D)]$$

- Learn to weight λ_T, λ_O, λ_U based on features

Variable Dependency Model
(Shi & Nie 2010)

- Discriminative model

$$P(Rel \mid Q, D) = \frac{1}{Z} \exp\left(\sum_{i}^{n} \lambda_i f_i(Q, D) \right)$$

$$\begin{aligned} P(Rel \mid Q, D) \propto & \sum_{q_i \in Q} \lambda_U(q_i \mid Q) f_U(q_i, D) \\ & + \sum_{q_i q_{i+1} \in Q} \lambda_B(q_i, q_{i+1} \mid Q) f_B(q_i q_{i+1}, D) \\ & + \sum_{w \in W} \sum_{q_i q_j \in Q, i \neq j} \lambda_{C_w}(q_i, q_j \mid Q) f_{C_w}(q_i, q_j, D) \end{aligned}$$

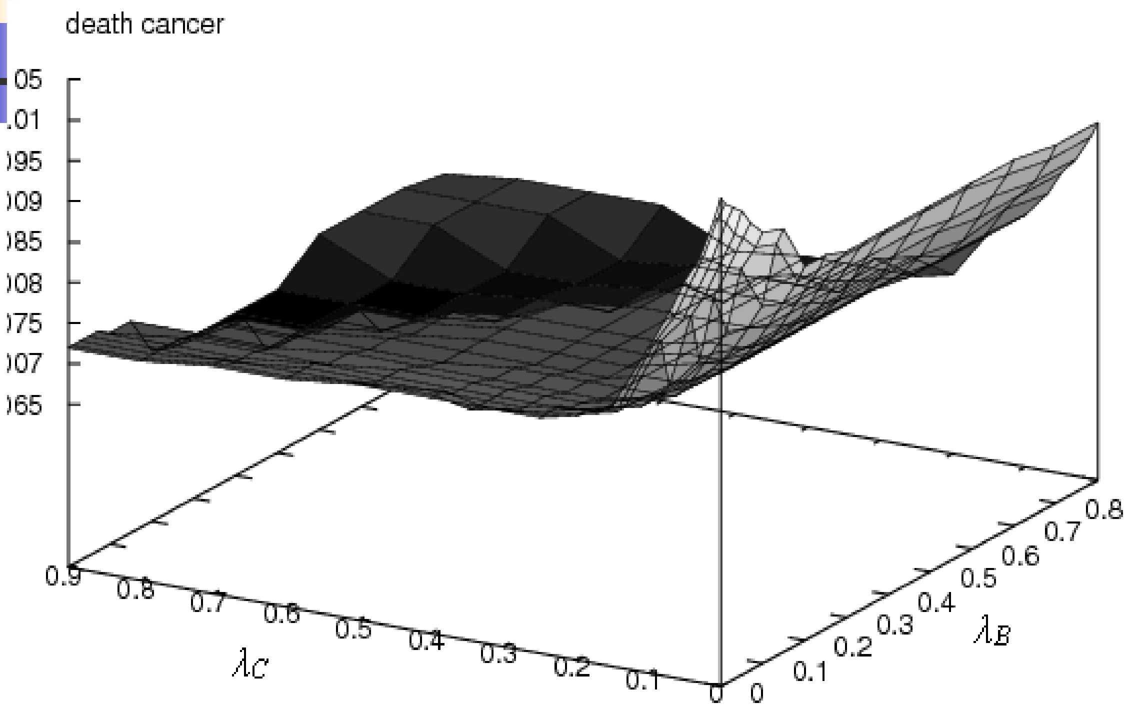
Variable Dependency Model (Shi & Nie 2010)

$$
\begin{aligned}
P(Rel \mid Q, D) \propto & \sum_{q_i \in Q} \lambda_U(q_i \mid Q) f_U(q_i, D) \\
& + \sum_{q_i q_{i+1} \in Q} \lambda_B(q_i, q_{i+1} \mid Q) f_B(q_i q_{i+1}, D) \\
& + \sum_{w \in W} \sum_{q_i q_j \in Q, i \neq j} \lambda_{C_w}(q_i, q_j \mid Q) f_{C_w}(q_i, q_j, D)
\end{aligned}
$$

- Unigram: $f_U = P_U(q_i \mid Q) \log P_U(q_i \mid D)$
- Ordered bigrams: $f_B = P_B(q_i q_{i+1} \mid Q) \log P_B(q_i q_{i+1} \mid D)$
- Unordered co-occurrence dependency within distance *w* (2, 4, 8): $f_{C_w} = P_{C_w}(\{q_i, q_j\} \mid Q) \log P_{C_w}(\{q_i, q_j\}_w \mid D)$

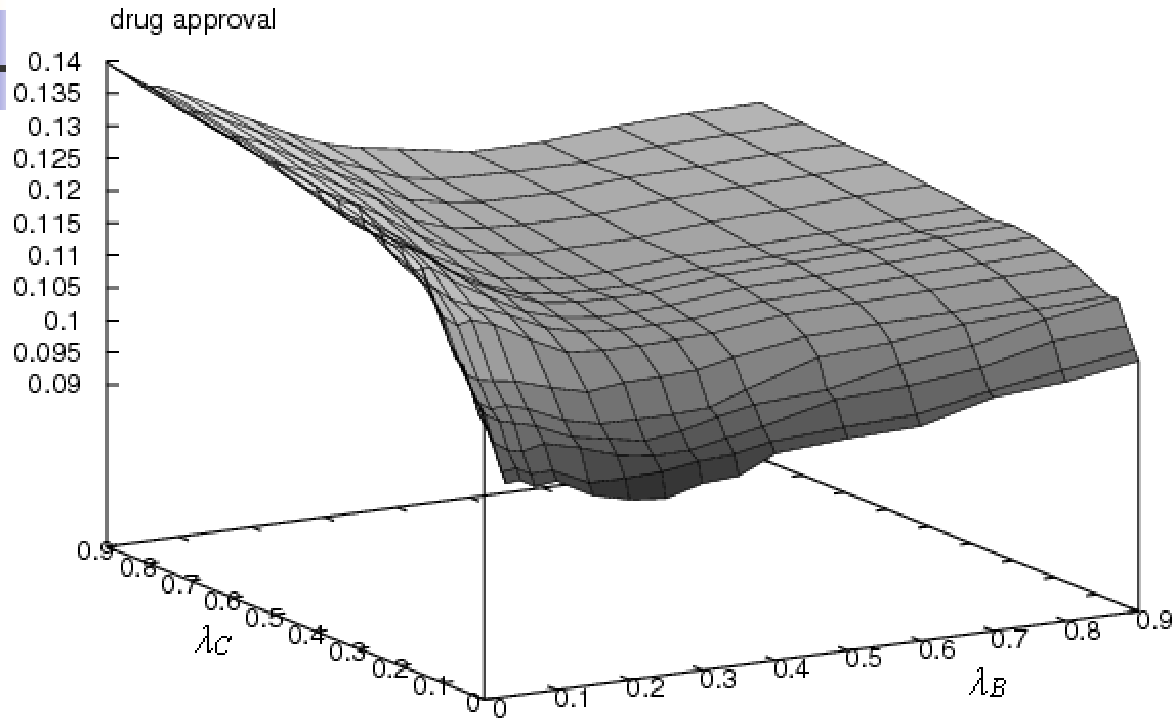
- λ_C, λ_B and λ_{Cw} are the importance of a particular term or dependency for the query
- Learning these parameters based on features

Death cancer – only unigrams



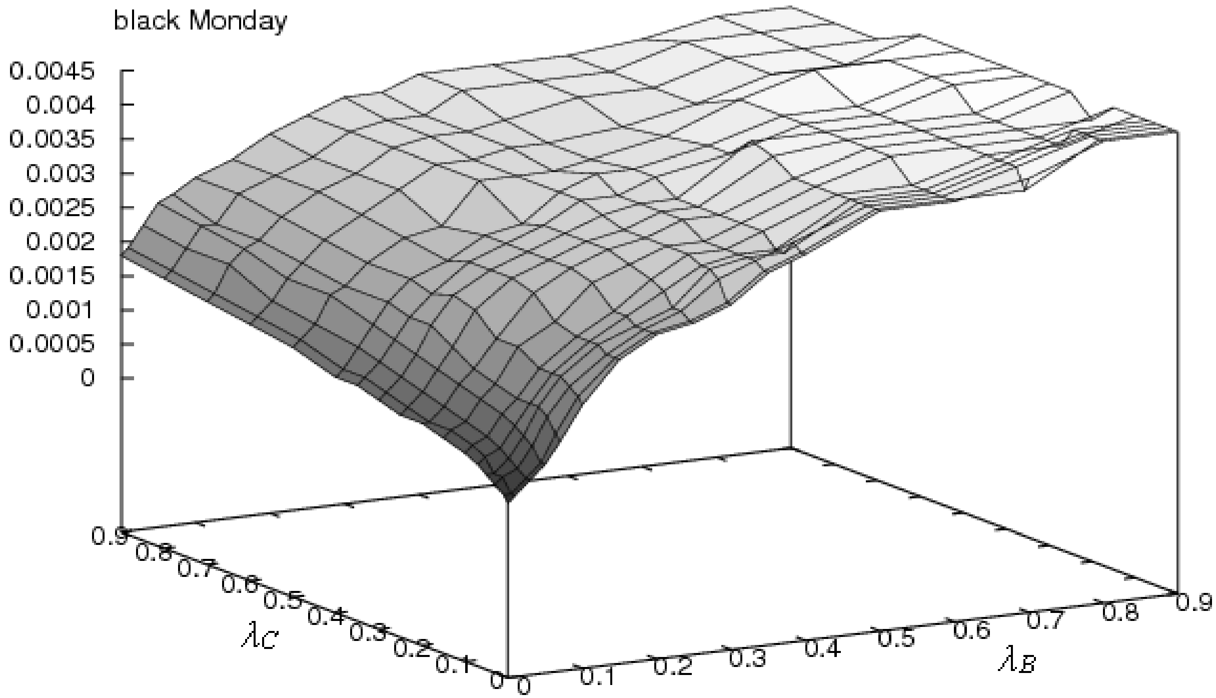
	MRF-SD	**DLM**	**Ideal**
weights	$.90U$ $.08B$ $.02C_8$	$.98U$ $.02B$	$1.0U$
MAP	0.0088	**0.0104**	0.0105

Drug approval – Use Co-occ. but not bigram



	MRF-SD	**DLM**	**Ideal**
weights	$.9U\ .08B\ .02C_8$	$.71U\ \ .29B$	$.40U\ .32B\ .28C_2$
MAP	0.0016	**0.0034**	0.0059

Black Monday – Co-occ and bigram



	MRF-SD	**DLM**	**Ideal**
weights	$.9U\ .08B\ .02C_8$	$.95U\ \ .02B$	$.28U\ .24C_4\ .24C_8\ .24C_{16}$
MAP	**0.1035**	0.0977	0.1505

Challenges

What has been achieved?

- Bag of words
  - Implemented as a vector space model, probabilistic model or language model
  - Achieve descent effectiveness
- Extensions
  - Term relations (programming→computer)
    - Relevance feedback
    - Thesauri
    - Co-occurrences in a corpus
    - Query logs (talk of Jianfeng Gao)
  - Dependency (computer – architecture)

Remaining problems

- We only considered the relevance between an isolated document-query pair
- Reality:
  - Documents can be connected (hyperlinks)
  - Queries can be related (query session)
  - Users may have typical behavior (interpret a query in similar ways)
- Recent efforts aim to consider these factors

From search to search intelligence

Information retrieval

- ~ Search engine
- Finding relevant information from a large set of documents
- User submits a query → search results

A review of generations of IR techniques

- Generation 1: basic models
- Generation 2: relations between documents, terms
- Generation 3: Learning from users
- Generation 4: Understanding users

Basic IR – Generation 1

- 1950s-1990s
- Extract keywords from documents
- Match query words against document keywords
- Document score using a manually defined function

Examples

- Document:

  e.g. “The area of information retrieval started in 1950s.”

  - Keywords: area, information, retrieval, started, 1950s
  - Some word processing: area, inform, retriev, start, 1950
- Query:
  - information retrieval
  - Inform, retriev
- Score: cosine similarity, BM25, language model

Characteristics of G1

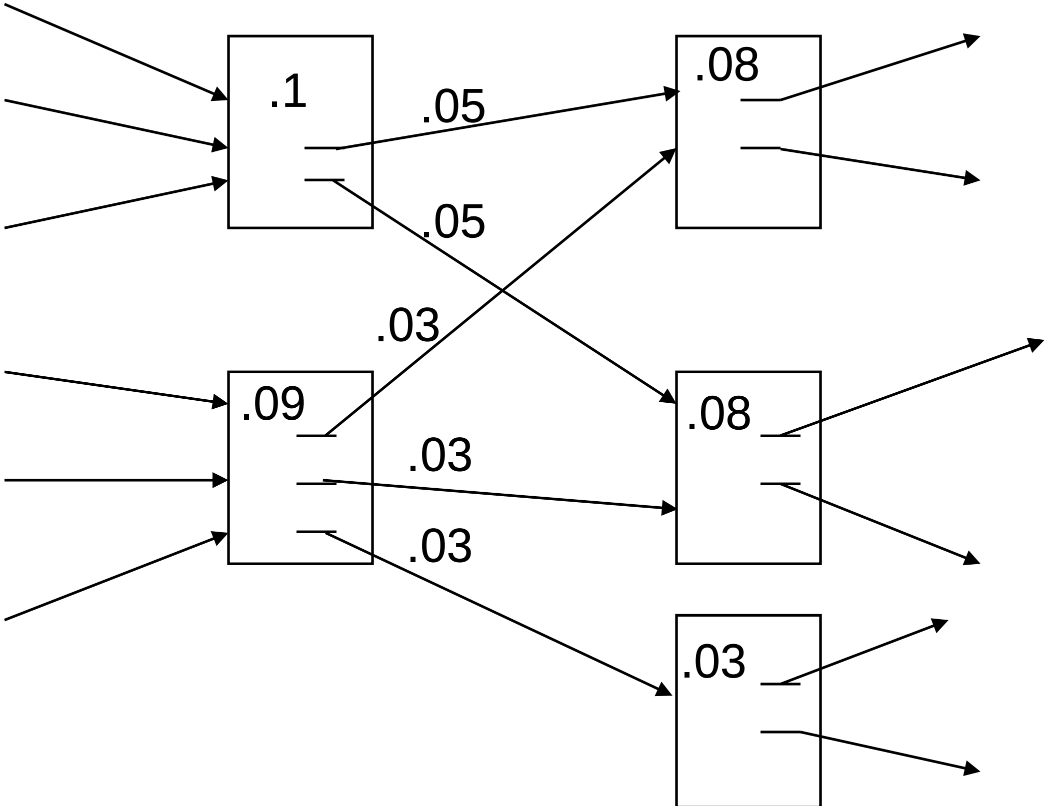
- Selection of meaningful words (stopword removal)
- Term weighting (tf*idf)
- Only content words (title, body, ...)
- Isolated documents
- Limited structure of documents
- Manually defined score functions
  - Cosine, BM25, probabilistic models, language models, ...

IR – Generation 2

- Started form ~1995
- Web search
- Connected documents (hyperlinks)
  - Anchor texts as additional description
  - Links as votes (popularity)
    - PageRank, HITS
- Links between terms (proximity)

PageRank – Basic idea

- Can view it as a process of PageRank "flowing" from pages to the pages they cite.



PageRank Algorithm

Let S be the total set of pages.

Let $\forall p \in S$: $E(p) = \alpha/|S|$ (for some $0<\alpha<1$, e.g. 0.15)

Initialize $\forall p \in S$: $R(p) = 1/|S|$

Until *convergence* (values do not change much) do

For each $p \in S$:

$$R'(p) = \sum_{q:q \to p} \frac{R(q)}{N_q} + E(p)$$

For each $p \in S$: $R(p) = cR'(p)$ (*normalize*)

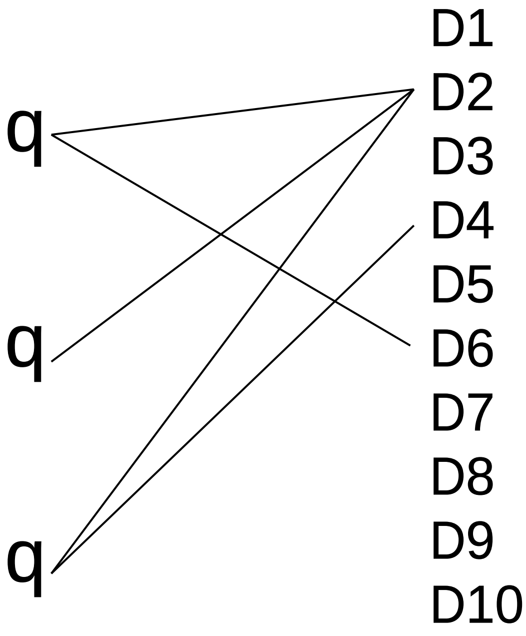
$$c = 1/\sum_{p \in S} R'(p)$$

Search engine – Generation 3

- Considering user interactions
  - Click-through: clicked documents are more relevant than unclicked ones
    - Noisy relevance judgments
  - Query sessions: queries of similar search intents
    - Relations between queries in the same session

Some ideas commonly used

- For a query, if many users clicked on a document, the document is likely relevant.



Some ideas commonly used

- A later query in the same session may be an alternative expression of the same intent
  - Used to suggest query rewriting
  - pdf reader
  - acrobat reader
  - free acrobat reader

User preference

Some ideas commonly used

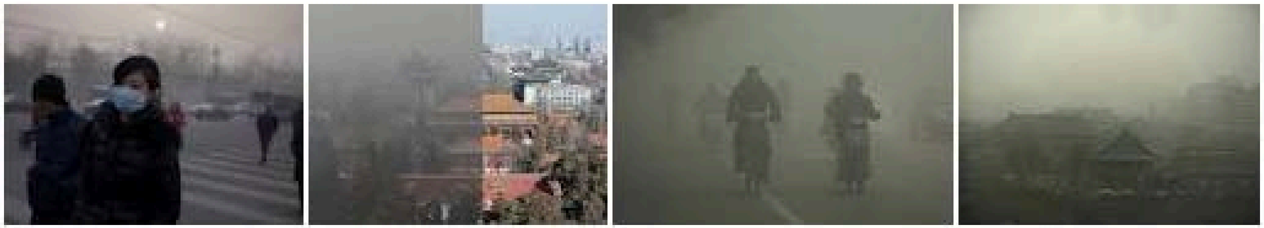
- Terms in a query are related to terms in the clicked documents (titles)
  - Query expansion

msn web	0.6675749
Webmensseger	0.6621253
msn online	0.6403270
windows web messanger	0.6321526
talking to friends on msn	0.6130790
school msn	0.5994550
msn anywhere	0.5667575
web message msn com	0.5476839
msn messager	0.5313351
hotmail web chat	0.5231608
messenger web version	0.5013624
instant messager msn	0.4550409

Clicked Title:
msn web messenger

An example of query expansion

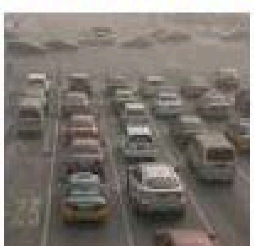
Images for **Beijing air pollution** Report images



More images for **Beijing air pollution**

Beijing Air Pollution: Real-time PM2.5 Air Quality Index (AQI)
aqicn.org/
Beijing AQI: **Beijing** Real-time **Air Quality** Index (AQI). ... **Beijing** PM25 (fine particulate matter) measured by **Beijing** Environmental Protection Monitoring Center (...
Beijing Air Pollution - Map - Shanghai - Central, Singapore Air Pollution

News for **Beijing air pollution**



China's Horrific **Air Quality** Prompts Its Government to ...
VICE News - 6 days ago
The policy probably won't do much to improve China's abysmal **air quality**, ... Highly polluted regions like the **Beijing**-Tianjin-Hebei

But may be wrong

Paper Cost

Web Images News Videos More Search tools

About 770,000,000 results (0.32 seconds)

Staples® Multipurpose Paper, 8 1/2" x 11", Ream | Make ...
www.staples.com › Paper & Stationery › Copy & Multipurpose Paper
Shop Staples® for Staples® Multipurpose **Paper**, 8 1/2" x 11", Ream and enjoy everyday low **prices**, plus FREE shipping on orders over $29.99. Get everything ...

Copy Paper / Printer Paper | Staples®
www.staples.ca/en/Copy-**Paper**...**Paper**/cat_CG2567_2-CA_1_20001
Whatever the project, having the right **paper** is key - whether you're looking for **cost**-effective general printer **paper** for basic print jobs or high quality color copy ...

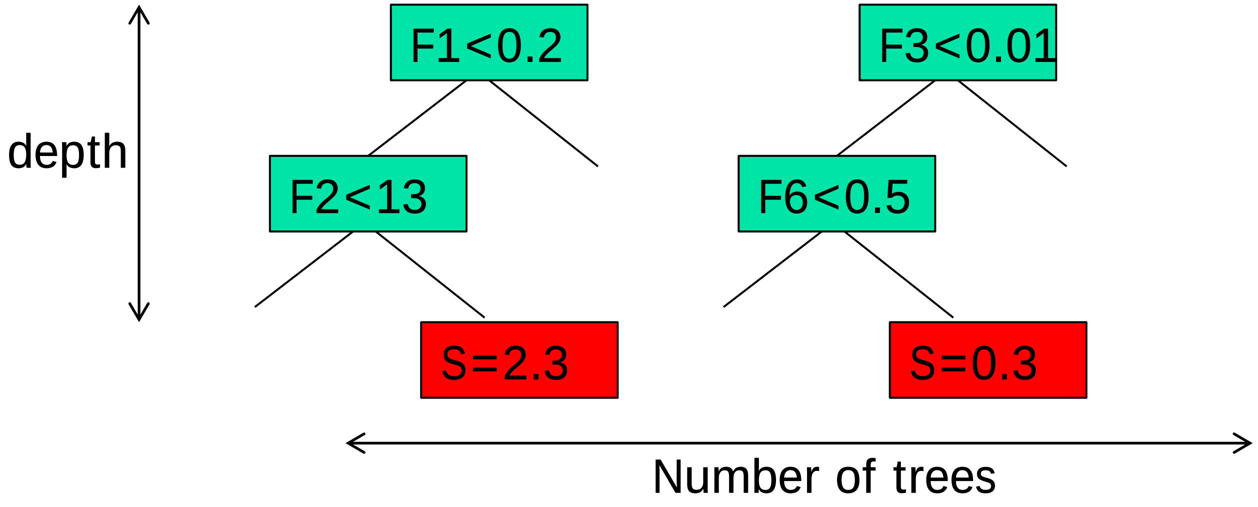
Paper Cost Calculator - CS Professional Suite from ...
https://cs.thomsonreuters.com/FileCabinetCS/worksheet.aspx
Paper Cost Calculator. A paperless software solution that saves you money. We' ve estimated that a typical firm with 500 clients could easily be spending more ...

Learning to rank – a major move

- It is difficult to define a ranking (score) function manually
- Learn the ranking function from users
  - Editorial judgments (relevance judgments by annotators)
    - Perfect(4), Very good (3), Good (2), Fair(1) Bad(0)
  - User clicks
    - Less accurate than editorial data, but there are much more
    - May reflect real user's true intent

Principles of learning-to-rank

- Regression (pointwise L2R): define a score function to fit the editorial judgments
  - E.g. Gradient Boosting Regression Trees
    - Editorial judgments for Q-D pairs
    - Extract features (F1, ...): e.g. tfidf, in-title?, clicked?, ...
    - Train a set of decision trees to fit the true scores



Principles of learning-to-rank

- Absolute relevance judgments are difficult to make
  - Inconsistency between annotators
- Easier to rank to documents
  - D1 is better than D2, i.e. $D1 \succ D2$
- The goal of a search engine is to rank, not to produce absolute relevance score
- Learn to rank documents

Principles of learning-to-rank

- Transform a set of editorial judgments to a set of preferences

  $$(Q, D_1) \succ (Q, D_2)$$

→ Good pairs (D_1,D_2) vs. Bad pairs (D_2,D_1)

- 2-class classification problem
  - RankSVM, ...
- Classify document pairs correctly
  - Pairwise L2R
- Or consider the whole order in a list of document
  - Listwise L2R

*see (Li 2011) and (Liu 2011) for details

Characteristics of generation 3

- No longer a manually defined ranking function
- Learn from users
- Flexible to incorporate various types of features on
  - Query
  - Document
  - Document-query pair
  - User’s search space / user profile
  - ...

Search engine – Generation 4

- Trend: search engines are evolving towards:
  - Understanding user’s search intent
    - Java: programming language, Java island or coffee?
    - Result diversification: mix up results for several possible intents in the first page

Query intent

- Query intent = a full specification of the documents the user wants to find
- A search query is partial specification of the information need and is not unique
- Often underspecified or ambiguous
  - 故宫
    - A general presentation?
    - A map or itinerary to go there?
    - A book?
    - …
  - Java transportation
    - Ambiguous query
- Intent mining: often based on query logs

Baidu百度 [新闻](#) 网页 [贴吧](#) [知道](#) [音乐](#) [图片](#) [视频](#) [地图](#) [文库](#) [更多»](#)

故宫 | 百度一下

故宫 携程 故宫博物院 20元起 贵就赔 随时退 piao.ctrip.com V3 推广链接

携程旅游-故宫门票特价1折起,凭身份证刷码入园,无需排队!随时退!贵就赔!

故宫_百度百科

故宫，世界文化遗产，全国重点文物保护单位，国家AAAAA级旅游景区。故宫旧称紫禁城。于明代永乐十八年（1420年）建成，建成后3月即失火烧毁，20年后重建。是明、...

其他含义:

- 南京故宫
- 沈阳故宫
- 查看"故宫"全部6个含义>>

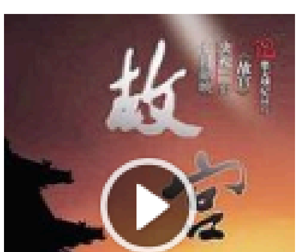
baike.baidu.com/ 2014-07-21

故宫博物院 官网

北京故宫博物院建立于1925年，是在明朝、清朝两代皇宫及其收藏的基础上建立起来的中国综合性博物馆，也是中国最大的古代文化艺术博物馆，其文物收藏主要来源于清代宫中旧藏。

www.dpm.org.cn/ 2014-05-19 V2 - 百度快照 - 评价

故宫 2005_全集高清视频在线观看_百度视频



更新至 2010-01-12

首播频道：CCTV

简介：12集《故宫》从故宫的建筑艺术、使用功能、馆藏文物和从皇宫到博物院的历程等方面，全面展示故宫辉煌瑰丽、神秘沧桑的宫殿建筑、丰富多彩，经历传奇的珍贵文物... 更多>>

How to do it?

- Add a diversity criterion to relevance in document ranking
- At each iteration, select a document that is both relevant to the query and different from the documents already selected

$$MMR(Q) = \underset{D_i \in R \setminus S}{\operatorname{argmax}} [\lambda Sim_1(D_i, Q) - (1-\lambda) \max_{D_j \in S} Sim_2(D_i, D_j)]$$

Some Extensions

- Try to cover the subtopics of the query as much as possible
  - xQuad (Santos et al. 2010, ...)
- Expand the initial query so as to have a more diversified pool
  - Diversified query expansion (Bouchoucha et al. 2012, 2013)
- Intent mining
  - NTCIR intent

Answer user's questions rather search

- Question answering and personalization

Knows where I am

A direct answer

新闻 网页 贴吧 知道 音乐 图片 视频 地图 文库 更多»
刘德华老婆的丈夫的老婆
百度一下
刘德华老婆的丈夫的老婆：
朱丽倩
朱丽倩英文名字叫carol，1966年4月6日出生在马来西亚槟城，有5个兄弟姐妹，家境富裕，父亲做地产和酒楼生意。1984年她和姐妹参加当地的"新潮小姐"选美获得季军，之后赴... 详情>>
来自百度百科 | 报错
推理过程：
刘德华
妻子
朱丽倩
丈夫
刘德华
妻子
朱丽倩

How to do it?

- Extract facts from reliable sources (Wikipedia, Baidu Zhidao, Baidu Baike, …)
- Build a knowledge graph
- Typical approach - IBM Watson Deep QA system
  - Given a question:
    - Can be answered by facts stored in knowledge graph?
    - Look for answers in free texts (passage retrieval + answer identification)
- Remaining issues
  - Non-factoid questions
  - More complex questions

Data analytics – Big data

- Answers to questions may be opinions (Baidu):
  - 蚕丝被能不能晒?
  - 孕妇能不能吃冰淇淋?
- Extract opinions from user answers
- Clustering opinions and display

Opinions

蚕丝被,纯正蚕丝被,哪家好? V1 - 推广

答：浦锦佳人蚕丝被,做工精细,100%品质保证,100%放心选购,值得信赖!产品质量投保!公司产品种类多,设计融贯中西,以“雅致,高贵”为基调,是馈赠亲朋,商务往来的品质之选!

回答者: 安徽浦锦佳人家.. 2014-08-02

蚕丝被可以晒吗（64条网友回答）：

87% 可以 56人回答

13% 不可以 8人回答

认为“可以”的网友回答：

回答1： 可以晒，晒晒可以去潮气，紫外线还能杀菌，只有好处没害处。 查看详细>>

一面山樵夫 | 十二级 采纳率55% 2012-04-28

回答2： 可以！蚕丝是高陷化合物，内含多种氨基酸，可以再春夏、夏秋之际选阳光明媚的天气晒一下（这几天就可以找出这么一天吧），但是不要在夏季的阳光下暴晒，那个阳光温度太高。蚕丝被我晒过的，没有问题！ 查看详细>>

Big data – the microblog case

- A huge amount of user generated contents
- 4V: Volume, Velocity, Variety, Veracity
- Noisy (many irrelevant and useless posts)
- But contain useful information, if correctly mined

Event monitoring

- Manually define a profile for a kind of events (e.g. a set of keywords)
  - 恐怖, 犯罪, 袭击, 伤亡, 爆炸, 团伙, ...
- Or construct a profile using a set of training data
- Monitor microblogs for significant increases of posts relating to the profile
- → a possible event of this kind

An example of World Cup

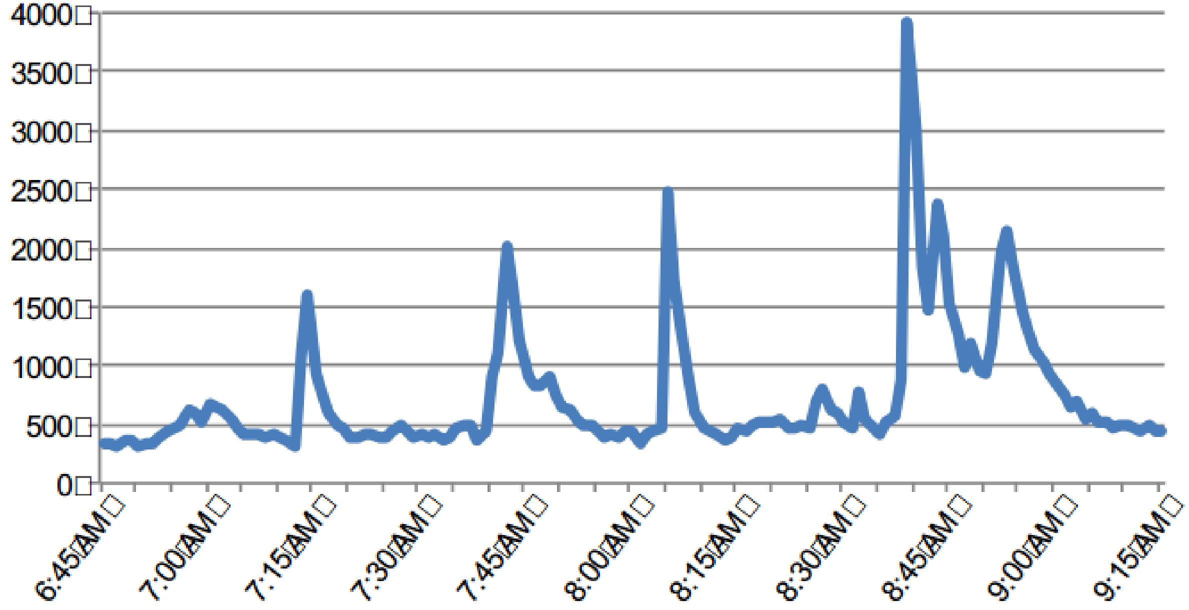


Figure 1. Twitter volume graph for the 2010 World Cup game of US vs. Slovenia. The x-axis is time and the y-axis is volume as measured in tweets/minute.

Earthquake

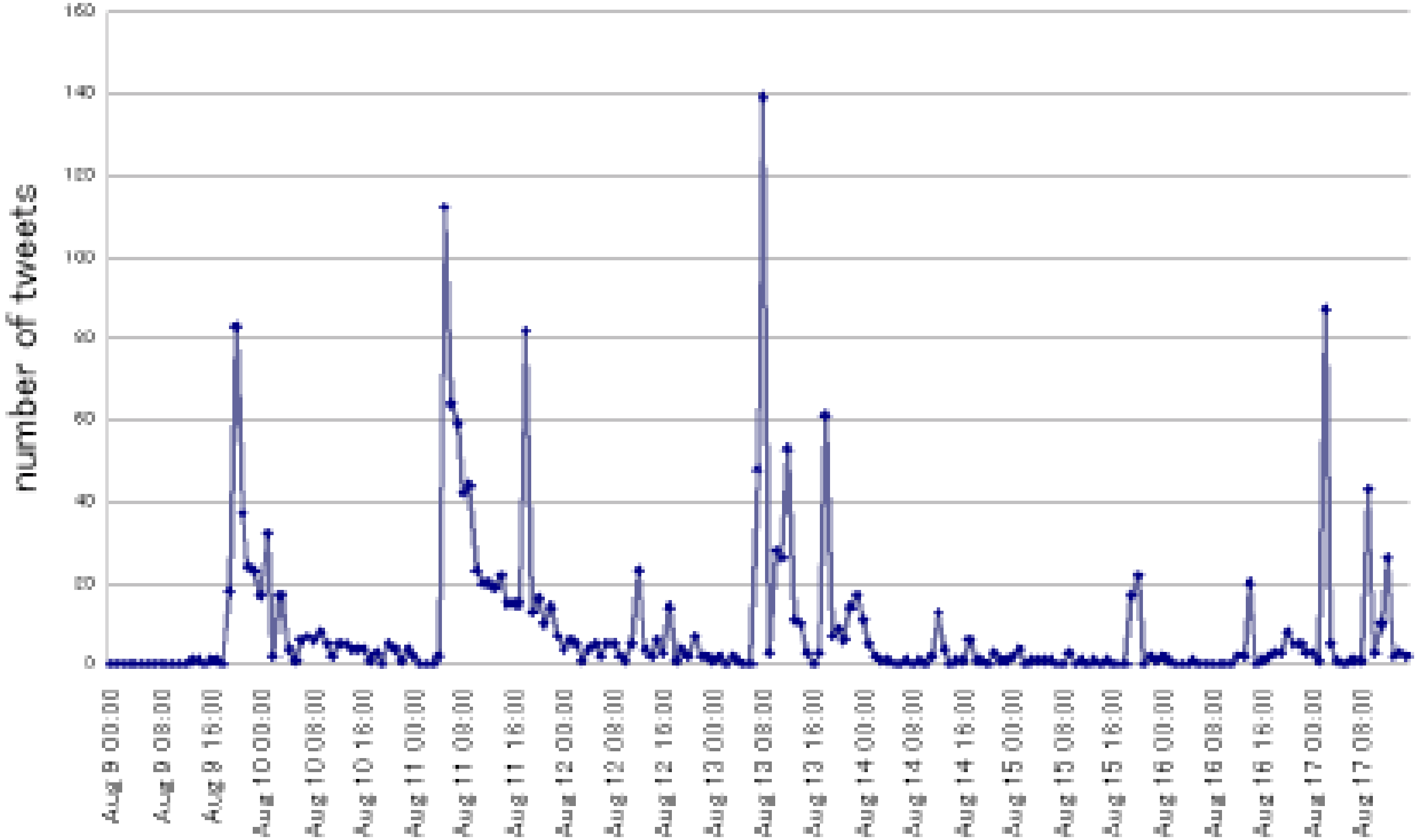
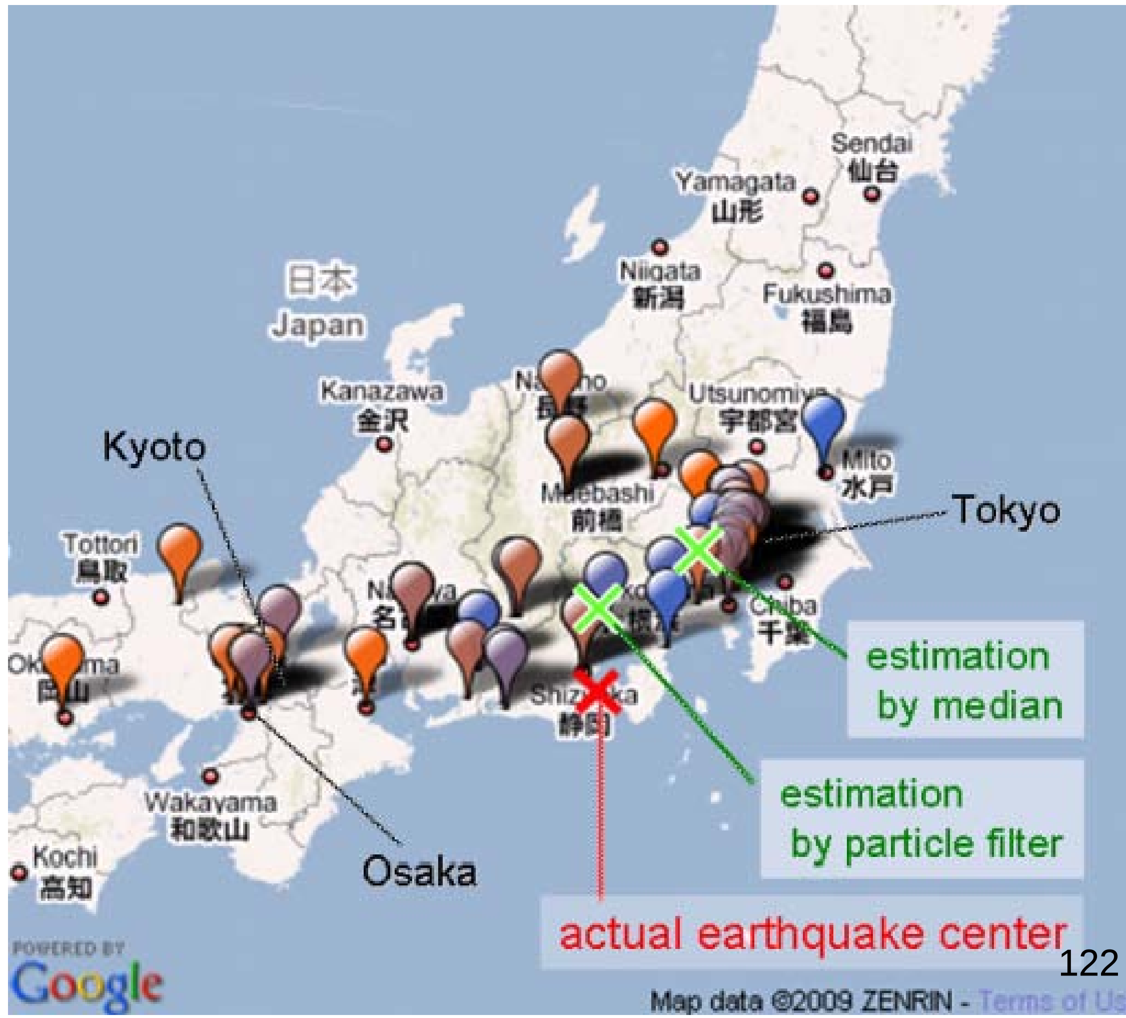


Figure 4: Number of tweets related to earthquakes.

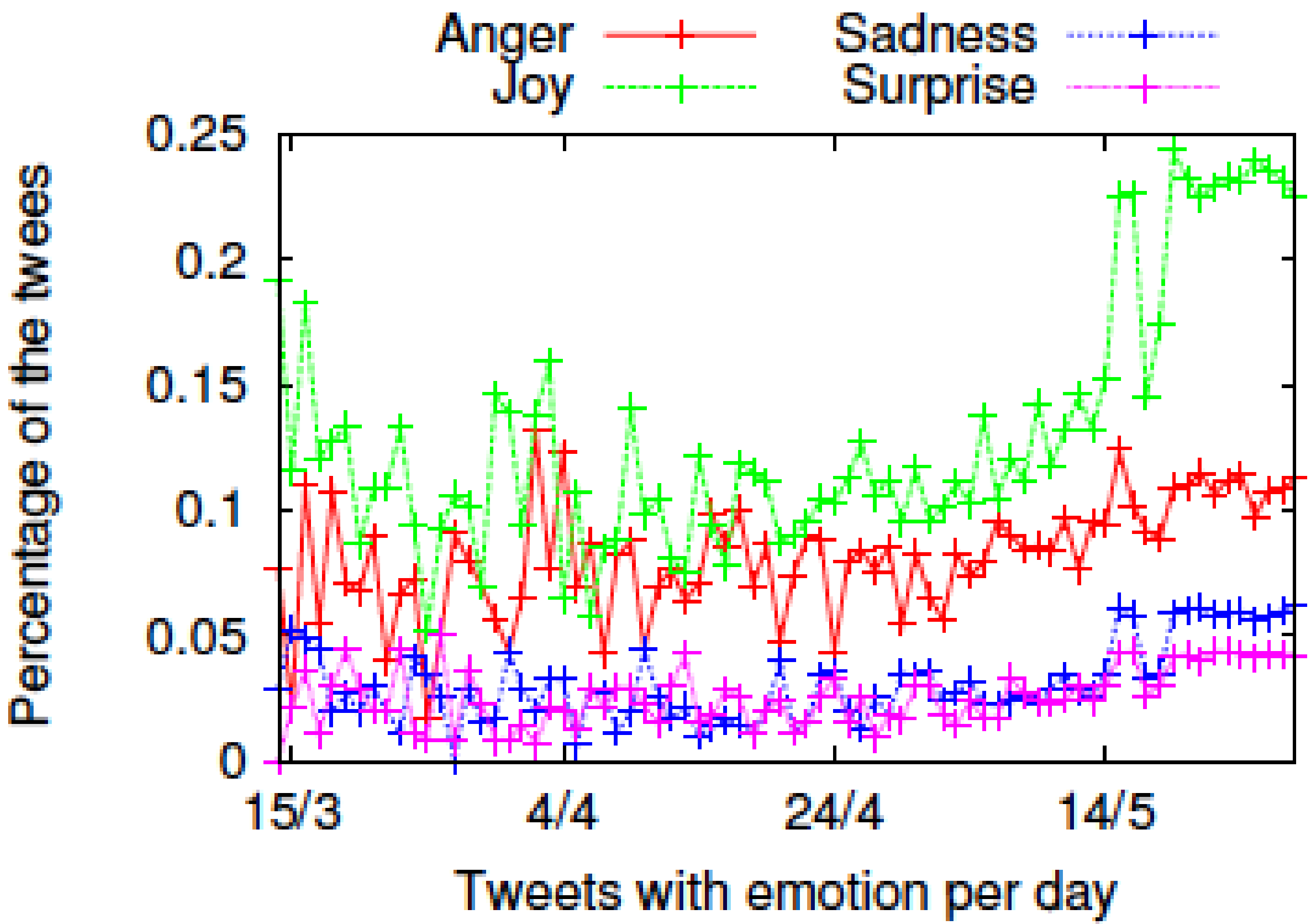
Event detection – Earthquake

Determine
where
the earthquake
is according to
the locations of
tweets:

- Twitter users
as sensors



Monitoring the mood of people



Some limitations

- Collect microblog posts using keywords?
- Data analytics

- Are the selected posts (data) really relevant?
- There is room for improvements:
  - => use more sophisticated IR techniques to select relevant information
  - => consider data reliability in data analysis

Final remarks

- Several generations of IR and search engines
  - Generation 1: Basic techniques
  - Generation 2: Link structure
  - Generation 3: User interactions
  - Generation 4: Understand users, diverse applications, including in Big data
- How far can we go?
  - Can we answer complex questions?
  - Can we do complex inference?
  - Can we satisfy diverse user needs (relevance -> user satisfaction)?
  - Can we determine relevant information in Big data?

Thanks
and
Questions

Traditional models

- Books:
  - Gerard Salton, Michael McGill, Introduction to modern information retrieval, McGraw-Hill, 1983 (classic)
  - Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.
  - Stefan Buettcher, Charles L. A. Clarke and Gordon V. Cormack, Information retrieval - Implementing and Evaluating Search Engines, MIT Press, 2010
  - W. Bruce Croft, Donald Metzler, Trevor Strohman, Search Engines: Information Retrieval in Practice, Pearson Education, 2009

Language modeling for IR

J.M. Ponte and W.B. Croft. 1998. A language modeling approach to information retrieval. In *SIGIR 21.*

D. Hiemstra. 1998. A linguistically motivated probabilistic model of information retrieval. *ECDL 2*, pp. 569–584.

A. Berger and J. Lafferty. 1999. Information retrieval as statistical translation. *SIGIR 22*, pp. 222–229.

Lavrenko, V. and Croft, W.B. Relevance-Based Language Models. In Proceedings of SIGIR Conf., pp. 120-127, 2001.

D.R.H. Miller, T. Leek, and R.M. Schwartz. 1999. A hidden Markov model information retrieval system. *SIGIR 22*, pp. 214–221.

Chengxiang Zhai, Statistical language models for information retrieval, in the series of Synthesis Lectures on Human Language Technologies, Morgan & Claypool, 2009

[Several relevant newer papers at *SIGIR* 2000–now.]

Workshop on Language Modeling and Information Retrieval, CMU 2001. http://la.lti.cs.cmu.edu/callan/Workshops/lmir01/ .

The Lemur Toolkit for Language Modeling and Information Retrieval. http://www-2.cs.cmu.edu/~lemur/ . CMU/Umass LM and IR system in C(++), currently actively developed.

Query/Doc. expansion

- Qiu, Y., and Frei, H.P. (1993). Concept query expansion. In *Proceedings of SIGIR Conf.*, pp. 160-169.
- Voorhees, E.M. (1994). Query Expansion Using Lexical-Semantic Relations In *Proceedings of SIGIR Conf.*, pp. 61-69.
- Cao, G., Nie, J.Y., and Bai, J. (2005). Integrating word relationships into language modeling. In *Proceedings of SIGIR Conf.*, pp. 298-305
- Bai, J. Nie, J., Bouchard, H. and Cao, G. (2007). Using query contexts in information retrieval. In *Proceedings of SIGIR Conf.*, pp. 15-22.

Dependence models

- Gao, J., Nie, J.-Y., Wu, G., and Cao, G. (2004). Dependence Language Model for Information Retrieval. In *Proceedings of the 2004 SIGIR Conf.,* pp. 170-177.
- Metzler, D. and Croft, W. B. (2005). A Markov random field model for term dependencies. In *Proceedings of SIGIR Conf.*, pp. 472-479
- M. Bendersky, D. Metzler and W. B. Croft: "Learning Concept Importance Using a Weighted Dependence Model" In Proceedings of WSDM 2010
- Lixin Shi, Jian-Yun Nie. Modeling Variable Dependencies between Characters in Chinese Information Retrieval, AIRS 2010

Learning to rank

- Fuhr, Norbert (1992), *Probabilistic Models in Information Retrieval*, *Computer Journal* **35** (3): 243–255, doi:10.1093/comjnl/35.3.243 (predecessor of L2R)
- Test data: Letorhttp://research.microsoft.com/en-us/um/beijing/projects/letor/

- Tie-Yan Liu (2009), *Learning to Rank for Information Retrieval*, *Foundations and Trends® in Information Retrieval*, Foundations and Trends in Information Retrieval: Vol. 3: No 3 **3** (3): 225–331
- Hang Li (2011), Learning to Rank for Information Retrieval and Natural Language Processing, Synthesis Lectures on Human Language Technologies, Morgan & Claypool, April 2011, 113 pages, (doi:10.2200/S00348ED1V01Y201104HLT012)

Result diversification and query intent mining

- Agrawal, R.; Gollapudi, S.; Halverson, A.; and Ieong, S. 2009. Diversifying search results. In Proc. of WSDM , 5–14.
- Carbonell, J., and Goldstein, J. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In Proc. of SIGIR , 335–336.
- Bouchoucha, A.; Liu, X.; and Nie, J.-Y. 2014. Integrating multiple resources for diversified query expansion. In Proc. of ECIR , 98–103.
- Dang, V., and Croft, W. B. 2012. Diversity by proportionality: An election-based approach to search result diversification. In Proc. of SIGIR, 65–74.
- Dang, V., and Croft, B. W. 2013. Term level search result diversification. In Proc. of SIGIR, 603–612.
- Xiaohua Liu, Arbi Bouchoucha, Alessandro Sordoni and Jian-Yun Nie, Compact Aspect Embedding For Diversified Query Expansions, AAAI, 2014
- Santos, R. L.; Macdonald, C.; and Ounis, I. 2010. Exploiting query reformulations for web search result diversification. In Proc. of WWW, 881–890.

- X Li, YY Wang, A Acero, Learning query intent from regularized click graphs, SIGIR’08, pp. 339-346
- J Hu, G Wang, F Lochovsky, J Sun, Z Chen, Understanding user's query intent with wikipedia, WWW’09, pp. 471-480
- F Radlinski, M Szummer, N Craswell, Inferring query intent from reformulations and clicks, WWW’10, pp. 1171-1172