Query Operations

Berlin Chen
Department of Computer Science & Information Engineering
National Taiwan Normal University

References:

1. *Modern Information Retrieval,* Chapter 5
2. *Introduction to Information Retrieval*, Chapter 9

Introduction

- Users have no detailed knowledge of
  - The collection makeup
  - The retrieval environment

Difficult to formulate queries

- Moreover, in most collections, the same concept may be referred to using different words
  - This issue, known as synonymy, has an impact on the recall of most IR systems

Scenario of (Web) IR

1. An initial (naive) query posed to retrieve relevant docs
2. Docs retrieved are examined for relevance and a new improved query formulation is constructed and posed again

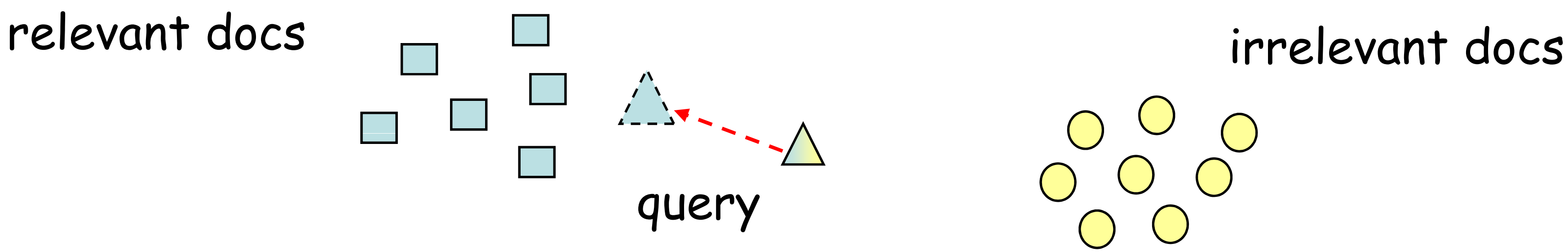
Expand the original query with new terms (**query expansion**) and rewight the terms in the expanded query (**term weighting**)

Query Reformulation

- Approaches through *query expansion* (QE) and *terming weighting*
  - Feedback information from the user
    - **Relevance feedback**
      - With vector, probabilistic models et al.
  - Information derived from the set of documents initially retrieved (called *local* set of documents)
    - **Local analysis**
      - Local clustering, local context analysis
  - Global information derived from document collection
    - **Global analysis**
      - Similar thesaurus or statistical thesaurus

Relevance Feedback

- User (or *Automatic*) Relevance Feedback
  - The most popular query reformation strategy
- Process for user relevance feedback
  - A list of retrieved docs is presented
  - User or system exam them (e.g. the top 10 or 20 docs) and marked the relevant ones
  - Important terms are selected from the docs marked as relevant, and the importance of them are enhanced in the new query formulation



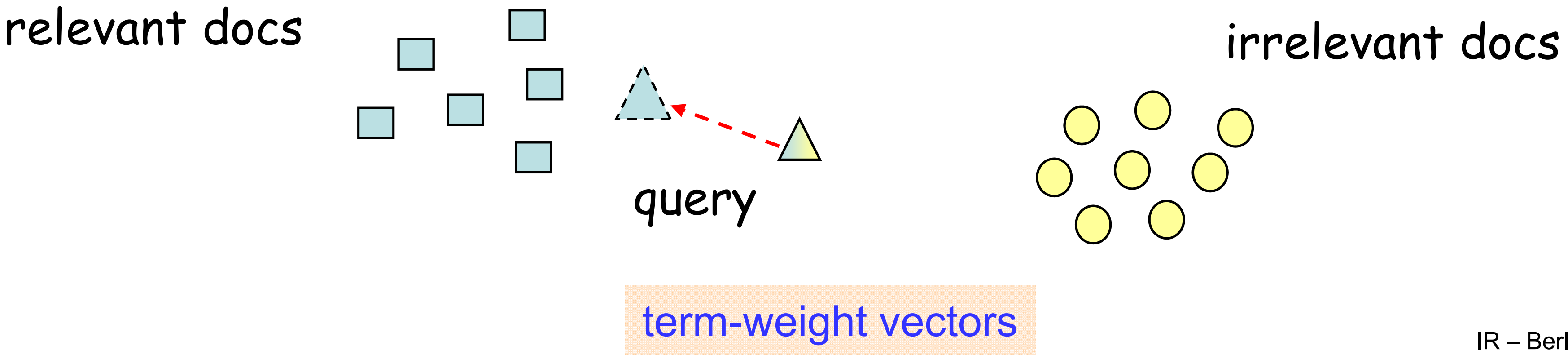
User Relevance Feedback

- Advantages
  - Shield users from details of query reformulation
    - User only have to provide a relevance judgment on docs
  - Break down the whole searching task into a sequence of small steps
  - Provide a controlled process designed to emphasize some terms (relevant ones) and de-emphasize others (non-relevant ones)

For **automatic relevance feedback**, the whole process is done in an implicit manner

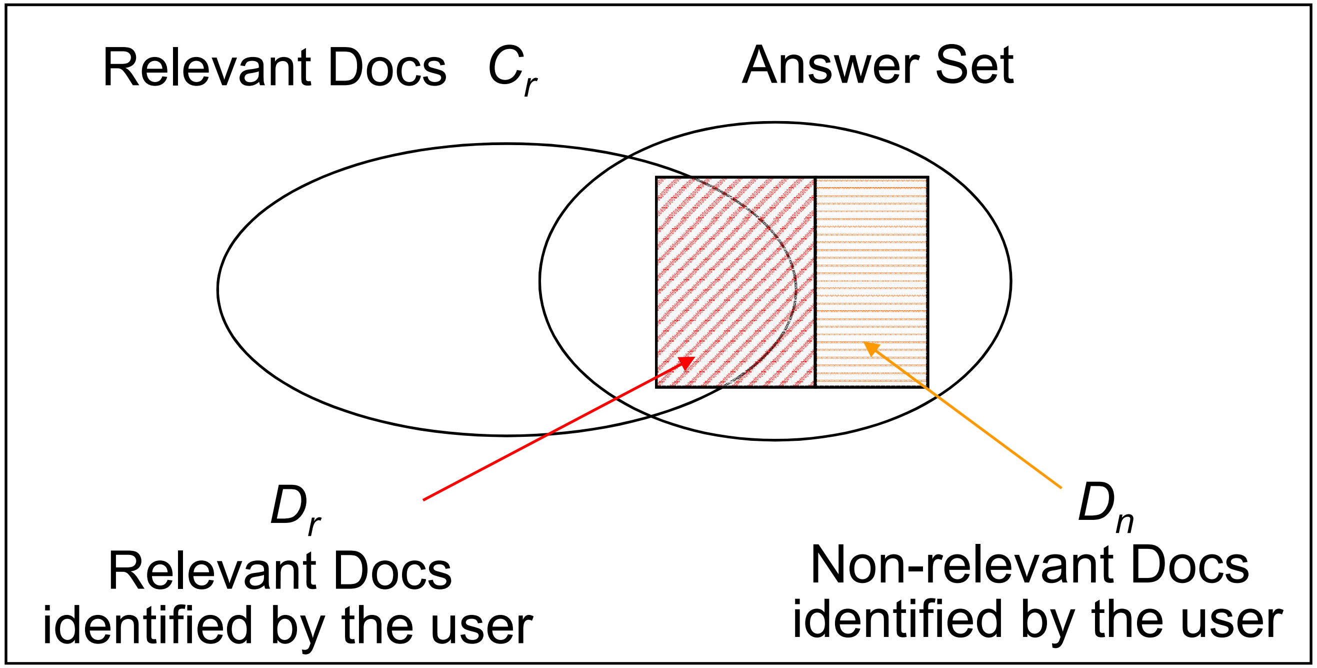
Query Expansion and Term Reweighting for the Vector Model

- **Assumptions**
  - Relevant docs have term-weight vectors that resemble each other
  - Non-relevant docs have term-weight vectors which are dissimilar from the ones for the relevant docs
  - The reformulated query gets to closer to the term-weight vector space of relevant docs



Query Expansion and Term Reweighting for the Vector Model (cont.)

- **Terminology**



Doc Collection with size N

Query Expansion and Term Reweighting for the Vector Model (cont.)

- **Optimal Condition**
  - The complete set of relevant docs C_r to a given query q is known in advance

$$\vec{q}_{opt} = \frac{1}{|C_r|} \sum_{\forall \vec{d}_i \in C_r} \vec{d}_i - \frac{1}{N - |C_r|} \sum_{\forall \vec{d}_j \notin C_r} \vec{d}_j$$

Elements in the final vector representation should be kept nonnegative (to be in the positive quadrant of the vector space)

  - Problem: the complete set of relevant docs C_r are not known a priori
    - **Solution**: formulate an initial query and incrementally change the initial query vector based on the known relevant/non-relevant docs
      - User or automatic judgments

Query Expansion and Term Reweighting for the Vector Model (cont.)

- **In Practice**

1. Standard_Rocchio

Rocchio 1965

$$\vec{q}_m = \alpha \cdot \vec{q} + \frac{\beta}{|D_r|} \cdot \sum_{\forall \vec{d}_i \in D_r} \vec{d}_i - \frac{\gamma}{|D_n|} \cdot \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

modified query

initial/original query

2. Ide_Regular

Elements in the final vector representation should be kept nonnegative (to be in the positive quadrant of the vector space)

$$\vec{q}_m = \alpha \cdot \vec{q} + \beta \cdot \sum_{\forall \vec{d}_i \in D_r} \vec{d}_i - \gamma \cdot \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

3. Ide_Dec_Hi

The highest ranked non-relevant doc

$$\vec{q}_m = \alpha \cdot \vec{q} + \beta \cdot \sum_{\forall \vec{d}_i \in D_r} \vec{d}_i - \gamma \cdot \max_{non-relevant}\left(\vec{d}_j\right)$$

Positive feedback turns out to be much more valuable than negative feedback.

Query Expansion and Term Reweighting for the Vector Model (cont.)

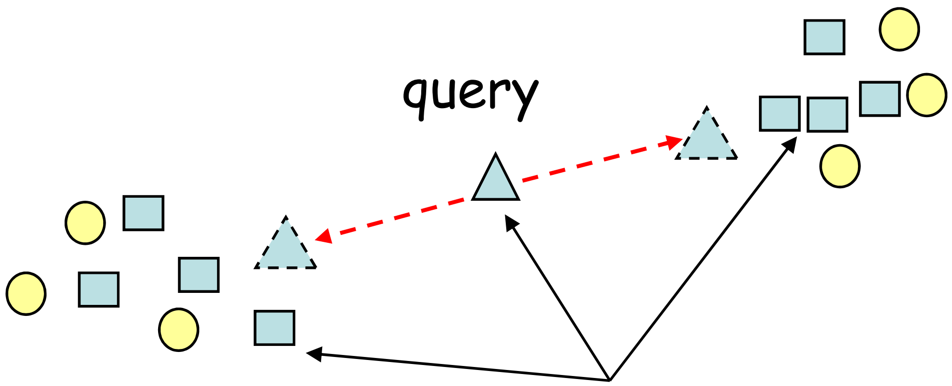
- **Some Observations**
  - Similar results were achieved for the above three approach (Dec-Hi slightly better in the past)
  - Usually, constant β is bigger than γ (why?)

- **In Practice** (cont.)
  - More about the constants
    - Rocchio, 1971: $\alpha=1$
    - Ide, 1971: $\alpha=\beta=\gamma=1$
    - **Positive feedback strategy**: $\gamma=0$

In implementation, terms occurring in the relevant or non-relevant documents can be used in toto or selectively to reweight/argument or be moved from the initial query.

More on Relevance Feedback

- Advantages
  - Simple, good results
    - Modified term weights are computed directly from the retrieved docs
- Disadvantages
  - No optimality criterion
    - Empirical and heuristic



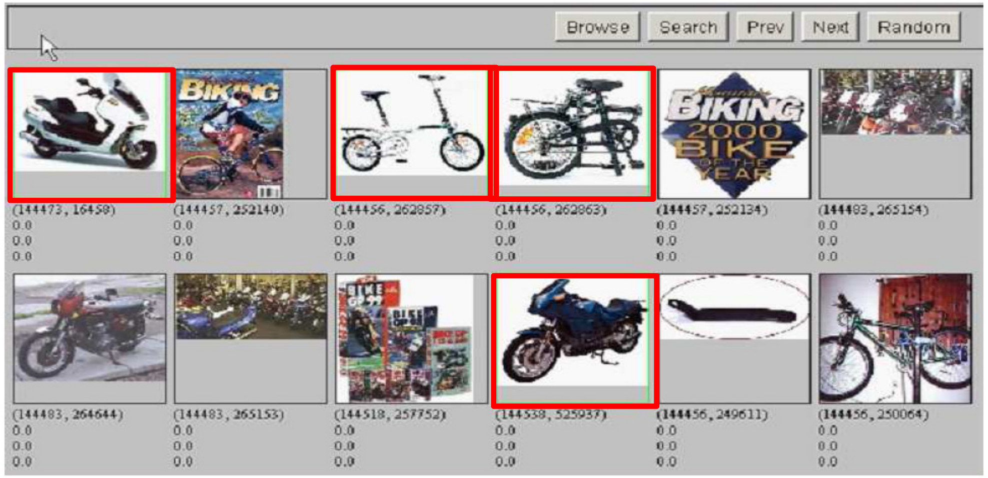
    (what if relevant documents belong to multiple clusters?)
  - High computing cost (potentially long response time)
    - Only reweight certain prominent terms in relevant docs?
  - There are still cases where relevance feedback alone is not sufficient: e.g., misspellings, cross-language IR, mismatch of searcher's versus collection vocabularies

More on Relevance Feedback (cont.)

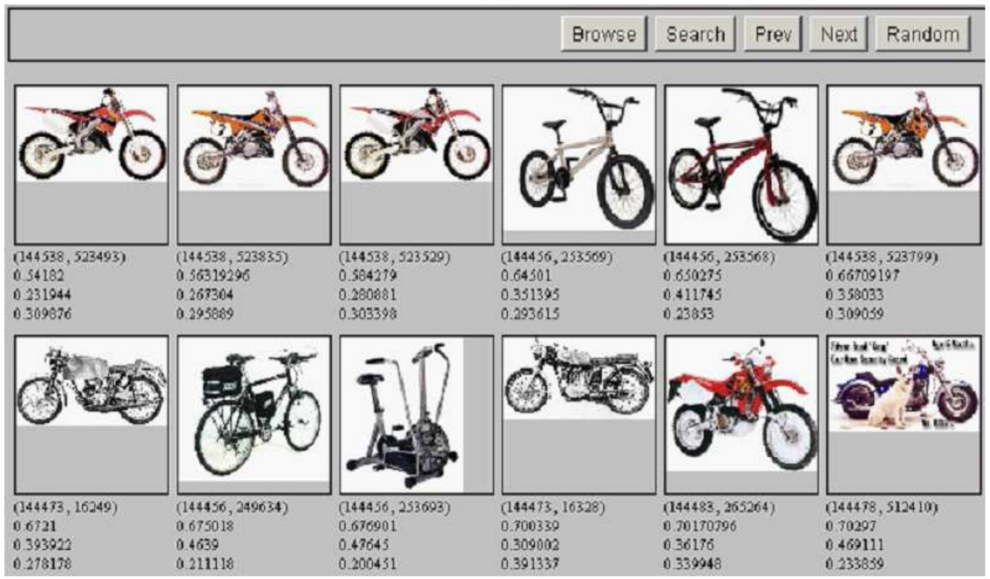
- Relevance feedback (QE+TW) has a side effect:
  - Tack a user's evolving information need
    - Seeing some documents may lead users to refine their understanding of the information they are seeking
- However, most Web search users would like to complete their search in a single interaction
    - Relevance feedback is mainly a recall enhancing strategy and Web search users are only rarely concerned with getting sufficient recall
    - An important more recent thread of work is the use of clickthrough data (through query log mining or clickstream mining) to provide indirect/implicit relevance feedback

Relevance Feedback for Image Search

The retrieved results with the initial text query"bike"



The new top-ranked results calculated after a round of relevance feedback



Term Reweighting for the Probabilistic Model

Roberston & Sparck Jones 1976

- **Similarity Measure**

$$sim(d_j,q) \approx \sum_{i=1}^{t} w_{i,q} \times w_{i,j} \times \left[\log \frac{P(k_i \mid R)}{1-\boxed{P(k_i \mid R)}} + \log \frac{1-P(k_i \mid \overline{R})}{P(k_i \mid \overline{R})} \right]$$

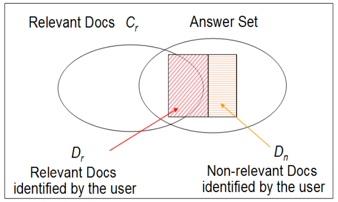
Binary weights (0 or 1) are used

prob. of observing term k_i in the set of relevant docs

- **Initial Search** (with some assumptions)
  - $P(k_i \mid R) = 0.5$:is constant for all indexing terms
  - $P(k_i \mid \overline{R}) = \frac{n_i}{N}$:approx. by doc freq. of index terms

$$\Rightarrow sim(d_j,q) \approx \sum_{i=1}^{t} w_{i,q} \times w_{i,j} \times \left[\log \frac{0.5}{1-0.5} + \log \frac{1-\frac{n_i}{N}}{\frac{n_i}{N}} \right]$$

$$= \boxed{\sum_{i=1}^{t} w_{i,q} \times w_{i,j} \times \log \frac{N-n_i}{n_i}}$$



Doc Collection with size N

Term Reweighting for the Probabilistic Model (cont.)

- **Relevance feedback** (term reweighting alone)

$$P(k_i \mid R) = \frac{|D_{r,i}|}{|D_r|}$$

← Relevant docs containing term k_i

← Relevant docs

$$P(k_i \mid \bar{R}) = \frac{n_i - |D_{r,i}|}{N - |D_r|}$$

Approach 1

$$P(k_i \mid R) = \frac{|D_{r,i}| + 0.5}{|D_r| + 1}$$

$$P(k_i \mid \bar{R}) = \frac{n_i - |D_{r,i}| + 0.5}{N - |D_r| + 1}$$

$$P(\bar{k}_i \mid R) = \frac{|D_r| - |D_{r,i}| + 0.5}{|D_r| + 1}$$

Approach 2

$$P(k_i \mid R) = \frac{|D_{r,i}| + \frac{n_i}{N}}{|D_r| + 1}$$

$$P(k_i \mid \bar{R}) = \frac{n_i - |D_{r,i}| + \frac{n_i}{N}}{N - |D_r| + 1}$$

Or $\frac{n_i - |D_{r,i}|}{N - |D_r|}$

$$sim(d_j, q) \approx \sum_{i=1}^{t} w_{i,q} \times w_{i,j} \times \left[\log \frac{\frac{|D_{r,i}|}{|D_r|}}{1 - \frac{|D_{r,i}|}{|D_r|}} + \log \frac{1 - \frac{n_i - |D_{r,i}|}{N - |D_r|}}{\frac{n_i - |D_{r,i}|}{N - |D_r|}} \right]$$

$$= \sum_{i=1}^{t} w_{i,q} \times w_{i,j} \times \log \left[\frac{|D_{r,i}|}{|D_r| - |D_{r,i}|} \cdot \frac{N - |D_r| - n_i + |D_{r,i}|}{n_i - |D_{r,i}|} \right]$$

Term Reweighting for the Probabilistic Model (cont.)

- Advantages
  - Feedback process is directly related to the derivation of new weights for query terms
  - The term reweighting is optimal under the assumptions of term independence and binary doc indexing
- Disadvantages
  - Document term weights are not taken into account
  - Weights of terms in previous query formulations are disregarded
  - No query expansion is used
    - The same set of index terms in the original query is reweighted over and over again

A Variant of Probabilistic Term Reweighting

Croft 1983

http://ciir.cs.umass.edu/

- **Differences**
  - Distinct initial search assumptions
  - Within-document frequency weight included
- **Initial search** (assumptions)

$$sim(d_j, q) \propto \sum_{i=1}^{t} w_{i,q} w_{i,j} F_{i,j,q}$$

$$F_{i,j,q} = (C + idf_i)\bar{f}_{i,j} \qquad \bar{f}_{i,j} = K + (1+K)\frac{f_{i,j}}{\max(f_{i,j})}$$

~ Inversed document frequency

~ Term frequency (normalized with the maximum within-document frequency)

- *C* and *K* are adjusted with respect to the doc collection

A Variant of Probabilistic Term Reweighting (cont.)

- **Relevance feedback**

$$F_{i,j,q} = \left(C + \log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)} + \log \frac{1 - P(k_i \mid \overline{R})}{P(k_i \mid \overline{R})}\right) \bar{f}_{i,j}$$

$$P(k_i \mid R) = \frac{|D_{r,i}| + 0.5}{|D_r| + 1}$$

$$P(k_i \mid \overline{R}) = \frac{n_i - |D_{r,i}| + 0.5}{N - |D_r| + 1}$$

A Variant of Probabilistic Term Reweighting (cont.)

- Advantages
  - The *within-doc frequencies* are considered
  - A normalized version of these frequencies is adopted
  - Constants *C* and *K* are introduced for greater flexibility

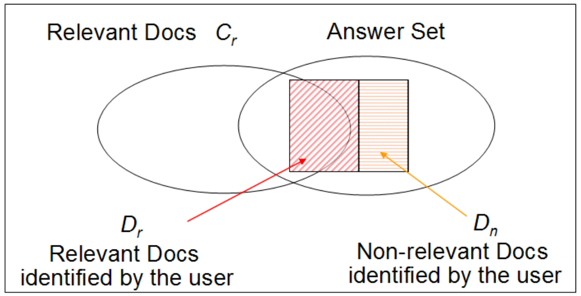
- Disadvantages
  - More complex formulation
  - No query expansion (just reweighting of index terms)

Evaluation of Relevance Feedback Strategies

- Recall-precision figures of user reference feedback is unrealistic
  - Since the user has seen the docs during reference feedback
    - A significant part of the improvement results from the higher ranks assigned to the set *R* of seen relevant docs

$$\vec{q}_m = \alpha \cdot \vec{q} + \frac{\beta}{|D_r|} \cdot \sum_{\forall \vec{d}_i \in Dr} \vec{d}_i - \frac{\gamma}{|D_n|} \cdot \sum_{\forall \vec{d}_j \in Dn} \vec{d}$$

modified query original query



Doc Collection with size *N*

  - The real gains in retrieval performance should be measured based on the docs not seen by the user yet

Evaluation of Relevance Feedback Strategies (cont.)

1. Recall-precision figures relative to the residual collection
   - The residual collection is the set of all docs minus the set of feedback docs provided by the user
   - Evaluate the retrieval performance of the modified query $\vec{q}_m$ considering only the residual collection
   - The recall-precision figures for $\vec{q}_m$ tend to be lower than the figures for the original query $\vec{q}$
     - It's OK ! If we just want to compare the performance of different relevance feedback strategies

Evaluation of Relevance Feedback Strategies (cont.)

2. Or alternatively, perform a comparative evaluation of $\vec{q}$ and $\vec{q}_m$ on another collection

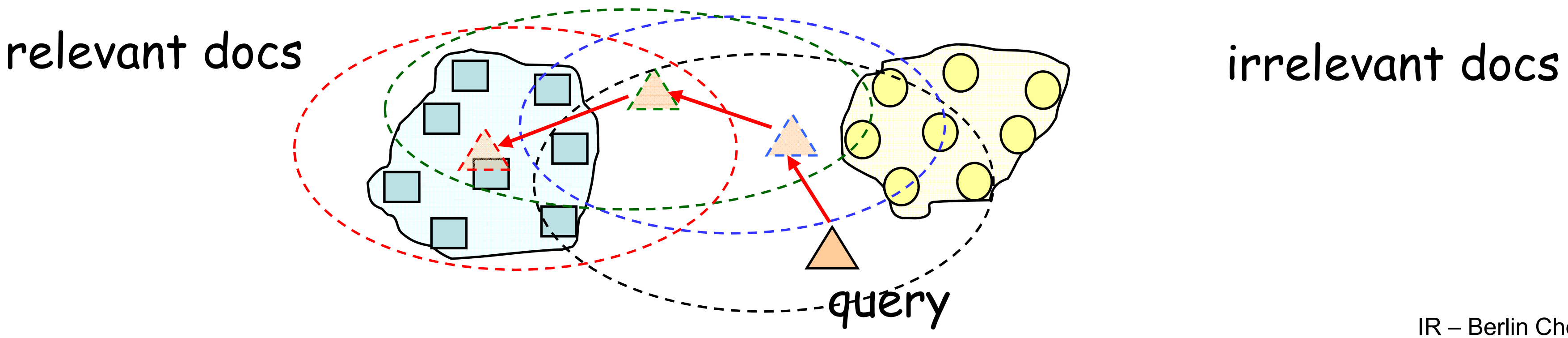
3. Or, the best evaluation of the utility of relevance feedback is to do user studies of its effectiveness in terms of how many documents a user find in a certain amount of time

Automatic Local/Global Analysis

- Remember that in user relevance feedback cycles
  - Top ranked docs separated into two classes
    - Relevant docs
    - Non-relevant docs
  - Terms in known relevant docs help describe a larger cluster of relevant docs
    - From a "**clustering**" perspective

Attar and Fraenkel 1977

  - Description of larger cluster of relevant docs is built iteratively **with assistance from the user**



Automatic Local/Global Analysis (cont.)

- *Alternative approach*: automatically obtain the description for a large cluster of relevant docs
  - Identify terms which are related to the query terms
    - Synonyms
    - Stemming variations
    - Terms are close each other in context

陳水扁 總統 李登輝 總統府 秘書長 陳師孟 一邊一國…

連戰 宋楚瑜 國民黨 一個中國 …

Automatic Local/Global Analysis (cont.)

- Two strategies
  - Global analysis
    - *All docs in collection* are used to determine a global *thesaurus-like* structure for QE
  - Local analysis
    - Similar to relevance feedback but without user interference
    - *Docs retrieved at query time* are used to determine terms for QE
    - *Local clustering, local context analysis*

QE through Local Clustering

- QE through **Clustering**
  - Build **global structures** such as **association matrices** to quantify term correlations
  - Use the correlated terms for QE
  - But not always effective in general collections

    陳水扁 總統 呂秀蓮 綠色矽島 勇哥 吳淑珍 …

    陳水扁 視察 阿里山 小火車
- QE through **Local Clustering**
  - Operate solely on the docs retrieved for the query
  - Not suitable for Web search: time consuming
  - Suitable for intranets
    - Especially, as the assistance for search information in specialized doc collections like medical (patent) doc collections

QE through Local Clustering (cont.)

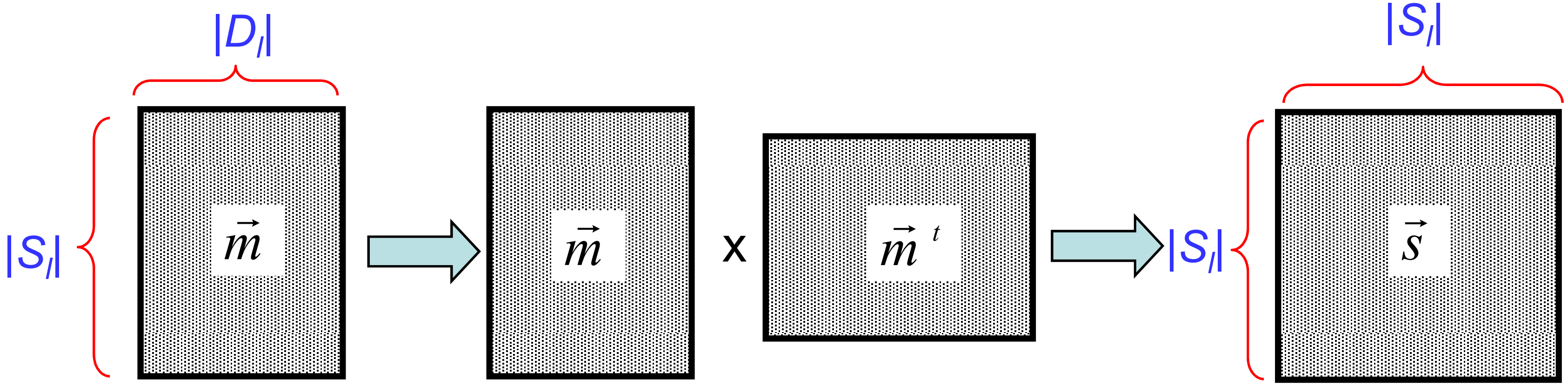
- Definition (Terminology)
  - Stem
    - $V(s)$: a non-empty subset of words which are grammatical variants of each other
      - E.g. {polish, polishing, polished}
    - A canonical form s of $V(s)$ is called a **stem**
      - e.g., s= polish
  - For a given query
    - Local doc set D_l : the set of documents retrieved
    - local vocabulary V_l : the set of all distinct words (stems) in the local document set
    - S_l: the set of all distinct stem derived from V_l

Strategies for Building Local Clusters

- **Association clusters**
  - Consider the co-occurrence of stems (terms) inside docs
- **Metric Clusters**
  - Consider the distance between two terms in a doc
- **Scalar Clusters**
  - Consider the neighborhoods of two terms
    - Do they have similar neighborhoods?

Strategies for Building Local Clusters (cont.)

- **Association clusters**
  - Based on the **co-occurrence** of stems (terms) inside docs
    - Assumption: stems co-occurring frequently inside docs have a **synonymity** association
  - An association matrix with $|S_l|$ rows and $|D_l|$ columns
    - Each entry $f_{s_i,j}$ the frequency of a stem s_i in a doc d_j



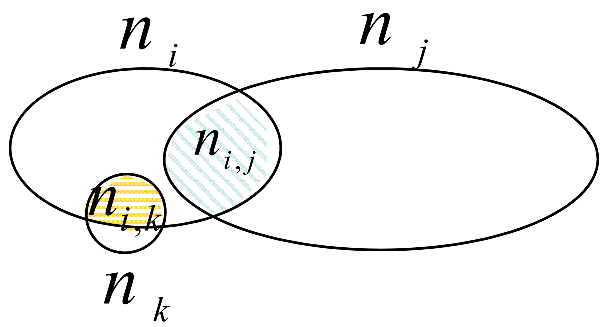
stem-doc matrix

stem-stem association matrix

Strategies for Building Local Clusters (cont.)

- **Association clusters**
  - Each entry in the stem-stem association matrix stands for **the correlation factor** between two stems

$$c_{u,v} = \sum_{d_j \in D_l} f_{s_u,j} \times f_{s_v,j}$$



  - The unnormalized form

$$s_{u,v} = c_{u,v}$$

    - Prefer terms with high frequency
  - The normalized form (ranged from 0 to 1)

$$s_{u,v} = \frac{c_{u,v}}{c_{u,u} + c_{v,v} - c_{u,v}}$$

Tanimoto coefficient

    - Prefer terms with low frequency

Strategies for Building Local Clusters (cont.)

- **Association clusters**
  - The u-th row in the association matrix stands all the associations for the stem s_u
  - A **local association cluster** $S_u(m)$
    - Defined as a set of stems s_v ($v \neq u$) with their respective values $s_{u,v}$ being the top m ones in the u-th row of the association matrix
  - Given a query, only the association clusters of query terms are calculated
    - The stems (terms) belong to the association clusters are selected and added the query formulation

Strategies for Building Local Clusters (cont.)

- **Association clusters**
  - Other measures for term association
    - Dice coefficient

$$s_{u,v} = \frac{2 \times c_{u,v}}{c_{u,u} + c_{v,v}}$$

    - Mutual information

$$s_{u,v} = MI\left(k_u, k_v\right) = \log \frac{P\left(k_u, k_v\right)}{P\left(k_u\right)P\left(k_v\right)} = \log \frac{\frac{n_{u,v}}{N}}{\frac{n_u}{N} \times \frac{n_v}{N}}$$

Strategies for Building Local Clusters (cont.)

- **Metric Clusters**
  - Key idea
    - Association clusters are simply based on the frequency of co-occurrence of pairs of terms in documents and do not take into account ***where*** the terms occur in a document
      - Two terms occurring in the same sentence seem more correlated than two terms occurring far apart in a document
    - It would be worthwhile to factor in the distance between two terms in the computation of their correlation factor

Strategies for Building Local Clusters (cont.)

- **Metric Clusters** $c_{u,v} \overset{?}{=} \sum_{d_j \in D_l} \sum_{k_i \in V(s_u)} \sum_{k_g \in V(s_v)} \frac{1}{r_j(k_i, k_g)}$
  - Take into consideration the distance between two terms in a doc while computing their correlation factor

$$c_{u,v} \overset{?}{=} \sum_{k_i \in V(s_u)} \sum_{k_g \in V(s_v)} \frac{1}{\boxed{r(k_i, k_g)}}$$

no. of words between k_i and k_g in the same doc

$r(k_i, k_g) = \infty$ if k_i and k_g are in distinct docs

  - The entry of **local stem-stem metric correlation** matrix $\vec{s}$ can be expressed as
    - The unnormalized form

$$s_{u,v} = c_{u,v}$$

    - The normalized form

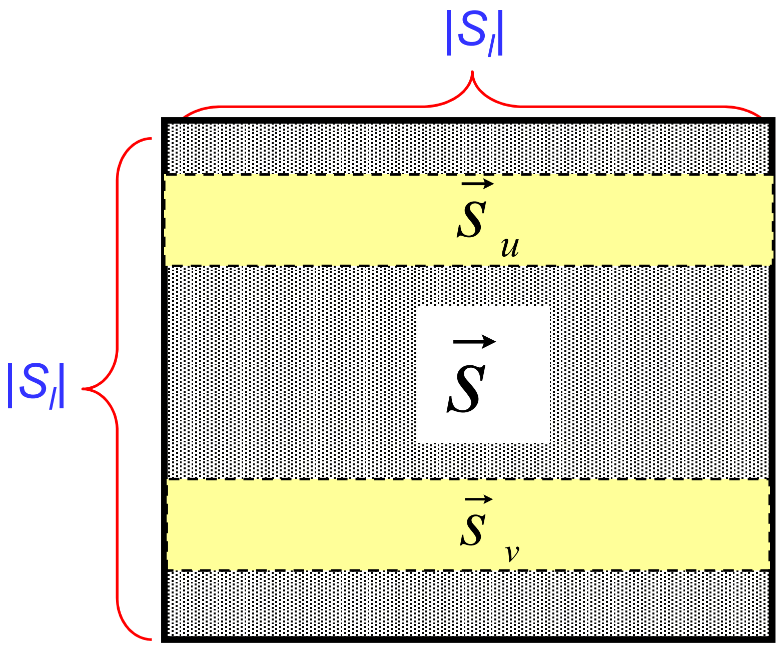
$$s_{u,v} = \frac{c_{u,v}}{|V(s_u)| \times |V(s_v)|}$$

ranged from 0 to 1

The local association clusters of stems can be similarly defined

Strategies for Building Local Clusters (cont.)

- **Scalar Clusters**
  - **Idea**: two stems (terms) with similar neighborhoods have some synonymity relationship
  - Derive the synonymity relationship between two stems by comparing the sets $S_u(m)$ and $S_v(m)$



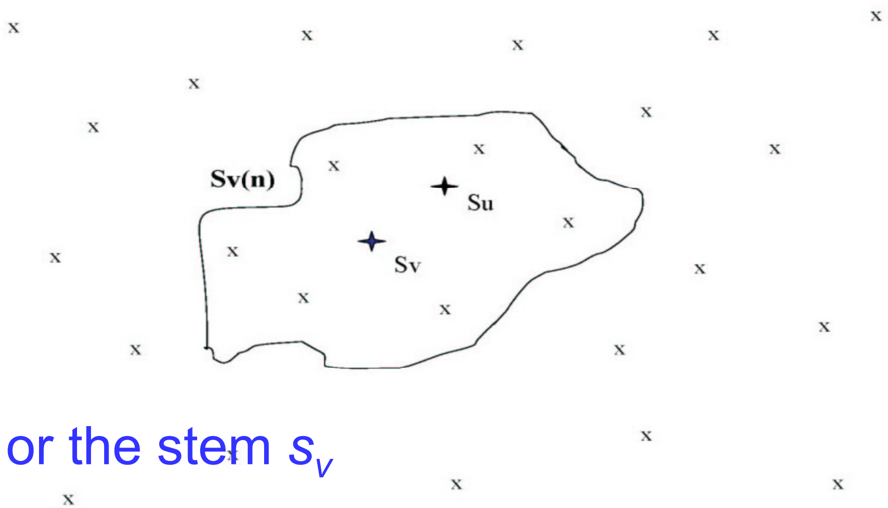
$$s_{u,v} = \frac{\vec{s}_u \cdot \vec{s}_v}{\left|\vec{s}_u\right| \times \left|\vec{s}_u\right|}$$

Use Cosine measure to derive a new scalar association matrix

The stem-stem association matrix achieved before

QE through Local Clustering (cont.)

- Iterative Search Formulation
  - "**neighbor**": a stem s_u belongs to a cluster associated to another term s_v is said to be a neighbor of s_v
    - Not necessarily synonyms in the grammatical sense
  - Stems belonging to clusters associated to the query stems (terms) can be used to expand the original query



stems s_u as a neighbor or the stem s_v

QE through Local Clustering (cont.)

- Iterative Search Formulation

  e.g, $s_{u,v} = \dfrac{c_{u,v}}{c_{u,u} + c_{v,v} - c_{u,v}}$

  – Query expansion
    - For each stem $s_v \in q$, select m neighbors stems from the cluster $S_v(m)$ and add them to the query
    - The additional neighbor stems will retrieve new relevant docs

  – The impact of normalized or unnormalized clusters
    - Unnormalized: group stems with high frequency
    - Normalized: group rare stems
    - Union of them provides a better representation of stem (term) correlations

Local Context Analysis

- Local Analysis
  - Based on the set of docs retrieved for the original query
  - Based on term (stem) correlation inside docs
  - Terms are neighbors of **each query terms** are used to expand the query

Calculation of term correlations at query time

- Global Analysis
  - Based on the whole doc collection
  - The thesaurus for term relationships are built by considering small contexts (e.g. passages) and phrase structures instead of the context of the whole doc
  - Terms closest to **the whole query** are selected for query expansion

Pre-calculation of term correlations

Local context analysis combines features from both

Local Context Analysis (cont.)

Xu and Croft 1996

- Operations of local context analysis
  - **Document concepts**: Noun groups (named concept here) from retrieved docs as the units for QE instead of single keywords
  - **Concepts** selected from the top ranked passages (instead of docs) based on their co-occurrence with the whole set of query terms (no stemming)

QE through Local Context Analysis

- The operations can be further described in three steps
  - Retrieve the top *n* ranked passages using the original query (a doc is segmented into several passages)
  - For each concept *c* in the top ranked passages, the similarity *sim*(*q,c*) between the whole query *q* and the concept *c* is computed using a variant of *tf-idf* ranking
  - The top *m* ranked concepts are added to the original query *q and* appropriately weighted, e.g.
    - Each concept is assigned a weight
      1-0.9x *i*/*m* (*i*: the position in rank)
    - Original query terms are stressed by a weight of 2

QE through Local Context Analysis (cont.)

- The similarity between a concept and a query

$$sim(q,c) = \prod_{k_i \in q} \left(\delta + \frac{\log\left(f(c,k_i) \times idf_c\right)}{\log n} \right)^{idf_i}$$

emphasize the infrequent terms

Set to 0.1 to avoid zero

the no. of top ranked passages considered

$$f(c,k_i) = \sum_{j=1}^{n} pf_{i,j} \times pf_{c,j}$$

Frequency of the concept c in passage j

$$idf_c = \max\left(1, \frac{\log_{10} N / np_c}{5}\right)$$

the no. of passages in the collection

$$idf_i = \max\left(1, \frac{\log_{10} N / np_i}{5}\right)$$

the no. of passages containing concept c

QE based on a Similarity Thesaurus

Qiu and Frei 1993

- Belongs to Global Analysis
- How to construct the similarity thesaurus
  - Term to term relationships rather than term co-occurrences are considered
- How to select term for query expansion
  - Terms for query expansion are selected based on their similarity to the whole query rather the similarities to individual terms

Docs are interpreted as indexing elements here

- Doc frequency within the term vector
- Inverse term frequency

N doc

t terms

$$\vec{k}_u = (w_{u,1}, w_{u,2}, \ldots, w_{u,N})$$

$$\vec{k}_v = (w_{v,1}, w_{v,2}, \ldots, w_{v,N})$$

term-doc matrix

QE based on a Similarity Thesaurus (cont.)

- Definition
  - $f_{u,j}$: the frequency of term k_u in document d_j
  - t_j : the number of distinct index terms in document d_j
  - Inverse term frequency

$$itf_j = \log \frac{t}{t_j}$$

(doc containing more distinct terms is less important)

- The weight associated with each entry in the term-doc matrix

$$w_{u,j} = \frac{\left(0.5 + 0.5 \frac{f_{u,j}}{\max_g f_{u,g}}\right) \times itf_j}{\sqrt{\sum_{l=1}^{N} \left[\left(0.5 + 0.5 \frac{f_{u,l}}{\max_g f_{u,g}}\right) \times itf_l\right]^2}}$$

The importance of the doc d_j to a term k_u

Let term vector have a unit norm

QE based on a Similarity Thesaurus (cont.)

- The relationship between two terms k_u and k_v

is just a cosine measure?

$$c_{u,v} = \vec{k}_u \cdot \vec{k}_v = \sum_{\forall d_j} w_{u,j} \times w_{v,j}$$

ranged from 0 to 1

  - The vector representations are normalized
  - The computation is computationally expensive
    - There may be several hundred thousands of docs

QE based on a Similarity Thesaurus (cont.)

Concept-based QE

- Steps for QE based on a similarity thesaurus
  1. Represent the query in the term-concept space

$$\vec{q} = \sum_{k_u \in q} w_{u,q} \times \vec{k}_u$$

  2. Based on the global thesaurus, compute a similarity between the each term k_v and the whole query q

$$sim(q, k_v) = \left(\sum_{k_u \in q} w_{u,q} \times \vec{k}_u \right) \cdot \vec{k}_v = \sum_{k_u \in q} w_{u,q} \times c_{u,v}$$

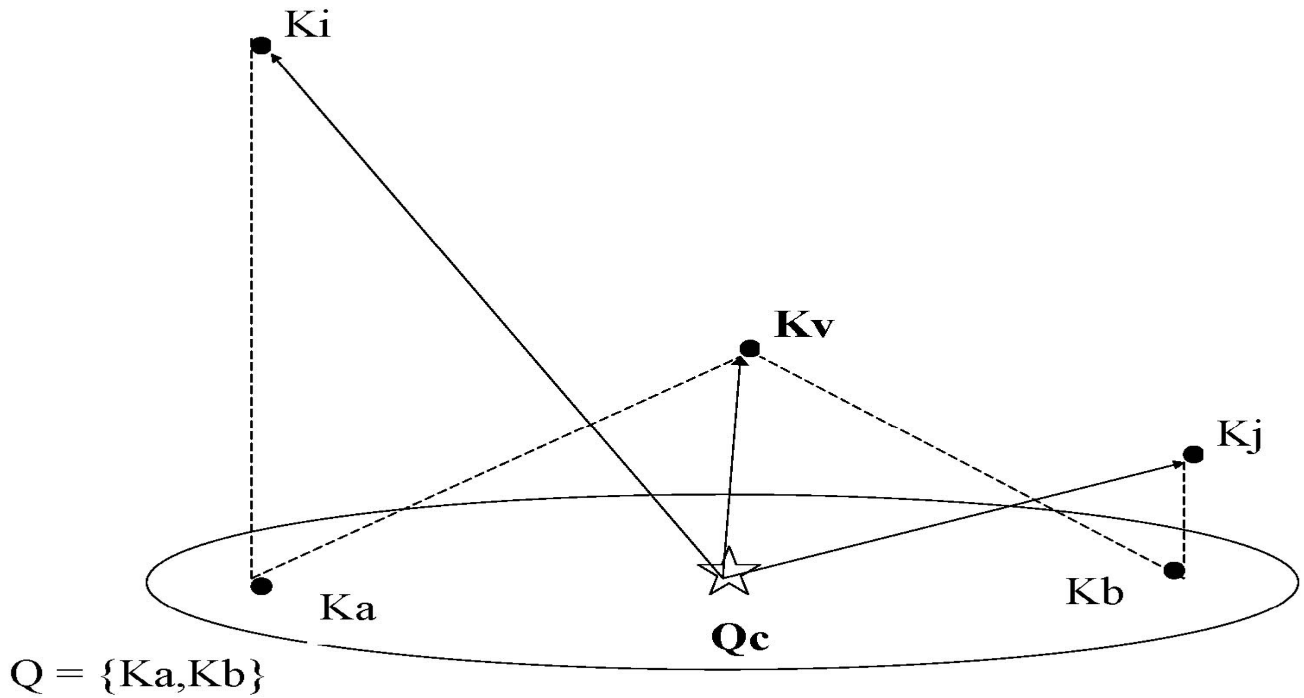
  3. Expand the query with the top r ranked terms according to $sim(q,k_v)$
     - The weight assigned to the expansion term

$$w_{v,q'} = \frac{sim(q, k_v)}{\sum_{k_u \in q} w_{u,q}} = \frac{\sum_{k_u \in q} w_{u,q} \times c_{u,v}}{\sum_{k_u \in q} w_{u,q}}$$

ranged from 0 to 1 ?

QE based on a Similarity Thesaurus (cont.)

- The term k_v selected for query expansion might be quite close to the whole query while its distances to individual query terms are larger



QE based on a Similarity Thesaurus (cont.)

- The similarity between query and doc measured in the term-concept space
  - Doc is first represented in the term-concept space

$$\vec{d}_j = \sum_{k_v \in d_j} w_{v,j} \times \vec{k}_v$$

  - Similarity measure

$$sim(q, d_j) \propto \sum_{k_v \in d_j} \sum_{k_u \in q} w_{v,j} \times w_{u,q} \times c_{u,v}$$

    - *Analogous to the formula for query-doc similarity in the generalized vector space model*
      - Differences
        - » Weight computation
        - » Only the top *r* ranked terms are used here

QE based on a Statistical Thesaurus

- Belongs to Global Analysis

- Global thesaurus is composed of classes which group correlated terms in the context of the whole collection

- Such correlated terms can then be used to expand the original user query
  - The terms selected must be **low frequency terms**
    - With high discrimination values

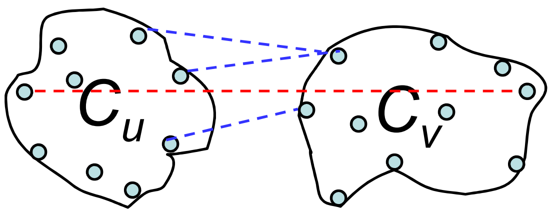
QE based on a Statistical Thesaurus (cont.)

- However, it is difficult to cluster **low frequency terms**
  - To circumvent this problem, we cluster docs into classes instead and use the low frequency terms in these docs to define our thesaurus classes
  - This algorithm must produce small and tight clusters
    - Depend on the cluster algorithm

QE based on a Statistical Thesaurus (cont.)

- Complete Link Algorithm
  - Place each doc in a distinct cluster
  - Compute the similarity between all pairs of clusters
  - Determine the pair of clusters $[C_u, C_v]$ with the highest inter-cluster similarity (using the cosine formula)
  - Merge the clusters C_u and C_v
  - Verify a stop criterion. If this criterion is not met then go back to step 2
  - Return a hierarchy of clusters

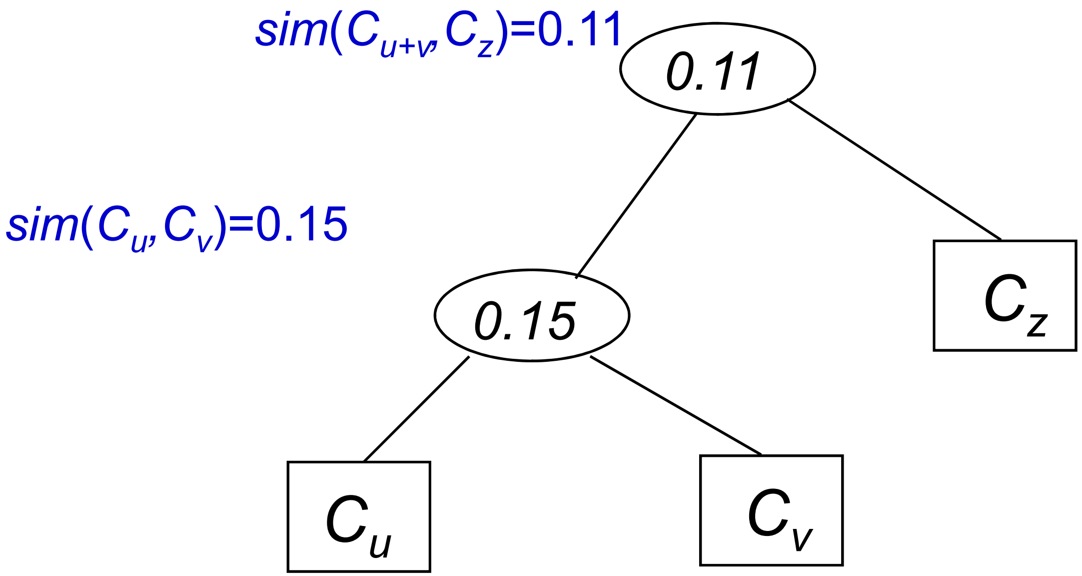
- Similarity between two clusters is defined as
  - The minimum of similarities between all pairs of inter-cluster docs



Cosine formula of the vector model is used

QE based on a Statistical Thesaurus (cont.)

- Example: hierarchy of three clusters



  - Higher level clusters represent a looser grouping
    - Similarities decrease as moving up in the hierarchy

QE based on a Statistical Thesaurus (cont.)

- Given the doc cluster hierarchy for the whole collection, the **terms** that compose each class of the global thesaurus are selected as follows
  - Three parameters obtained from the **user**
    - *TC*: Threshold class
    - *NDC*: Number of docs in class
    - *MIDF*: Minimum inverse doc frequency

QE based on a Statistical Thesaurus (cont.)

- Use the parameter *TC* as threshold value for determining the doc clusters that will be used to generate thesaurus classes
  - It has to be surpassed by sim(C_u,C_v) if the docs in the clusters C_u and C_v are to be selected as sources of terms for a thesaurus class

- Use the parameter *NDC* as a limit on the size of clusters (number of docs) to be considered
  - A low value of *NDC* might restrict the selection to the smaller clusters

QE based on a Statistical Thesaurus (cont.)

- Consider the set of docs in each doc cluster pre-selected above
  - Only the lower frequency terms are used as sources of terms for the thesaurus classes
  - The parameter *MIDF* defines the minimum value of inverse doc frequency for any term which is selected to participate in a thesaurus class

- Given the thesaurus classes have been built, they can be to query expansion

QE based on a Statistical Thesaurus (cont.)

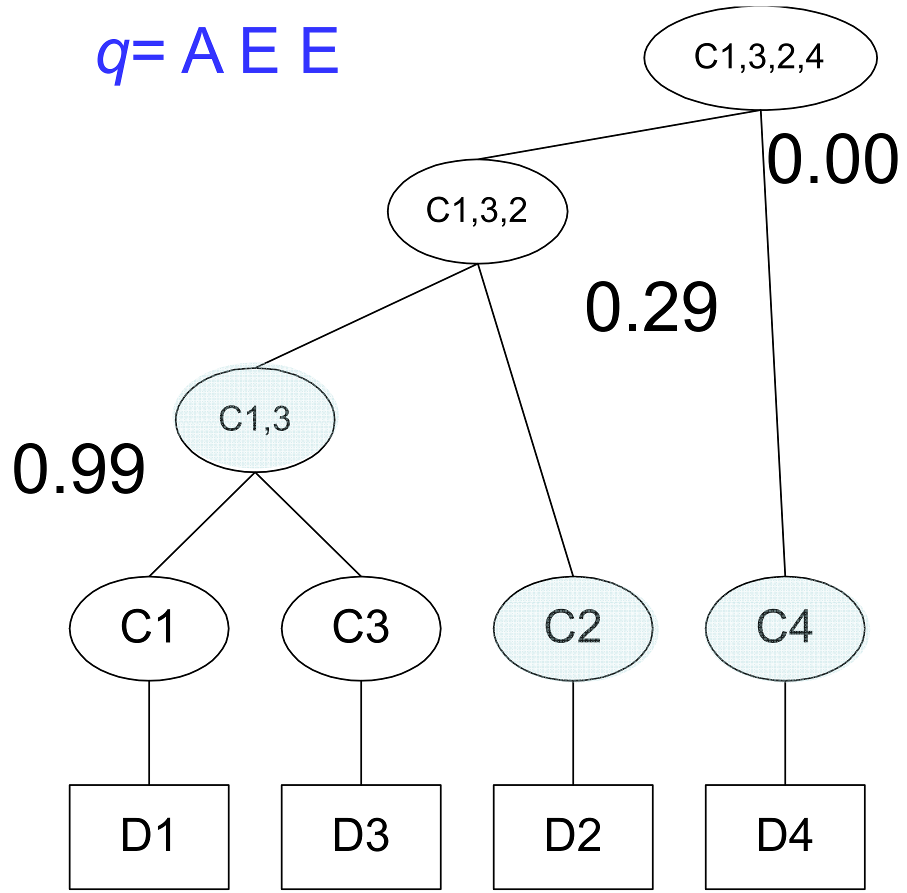
- Example

Doc1 = D, D, A, B, C, A, B, C
Doc2 = E, C, E, A, A, D
Doc3 = D, C, B, B, D, A, B, C, A
Doc4 = A

$$\left.\begin{array}{l} sim(1,3) = 0.99 \\ sim(1,2) = 0.40 \\ sim(2,3) = 0.29 \\ sim(4,1) = 0.00 \\ sim(4,2) = 0.00 \\ sim(4,3) = 0.00 \end{array}\right\} \text{Cosine formula with } \textit{tf-idf} \text{ weighting}$$

idf A = 0.0
idf B = 0.3
idf C = 0.12
idf D = 0.12
idf E = 0.60

q= A E E



- *TC* = 0.90 *NDC* = 2.00 *MIDF* = 0.2

q'=A B E E

QE based on a Statistical Thesaurus (cont.)

- Problems
  - Initialization of parameters *TC*, *NDC* and *MIDF*
  - *TC* depends on the collection
  - Inspection of the cluster hierarchy is almost always necessary for assisting with the setting of *TC*
  - A high value of *TC* might yield classes with too few terms
    - While a low value of *TC* yields too few classes

Trends and Research Issues (1/3)

- Visual display
  - Graphical interfaces (2D or 3D) for relevance feedback
    - Quickly identify (by visual inspection) relationships among doc in the answer set

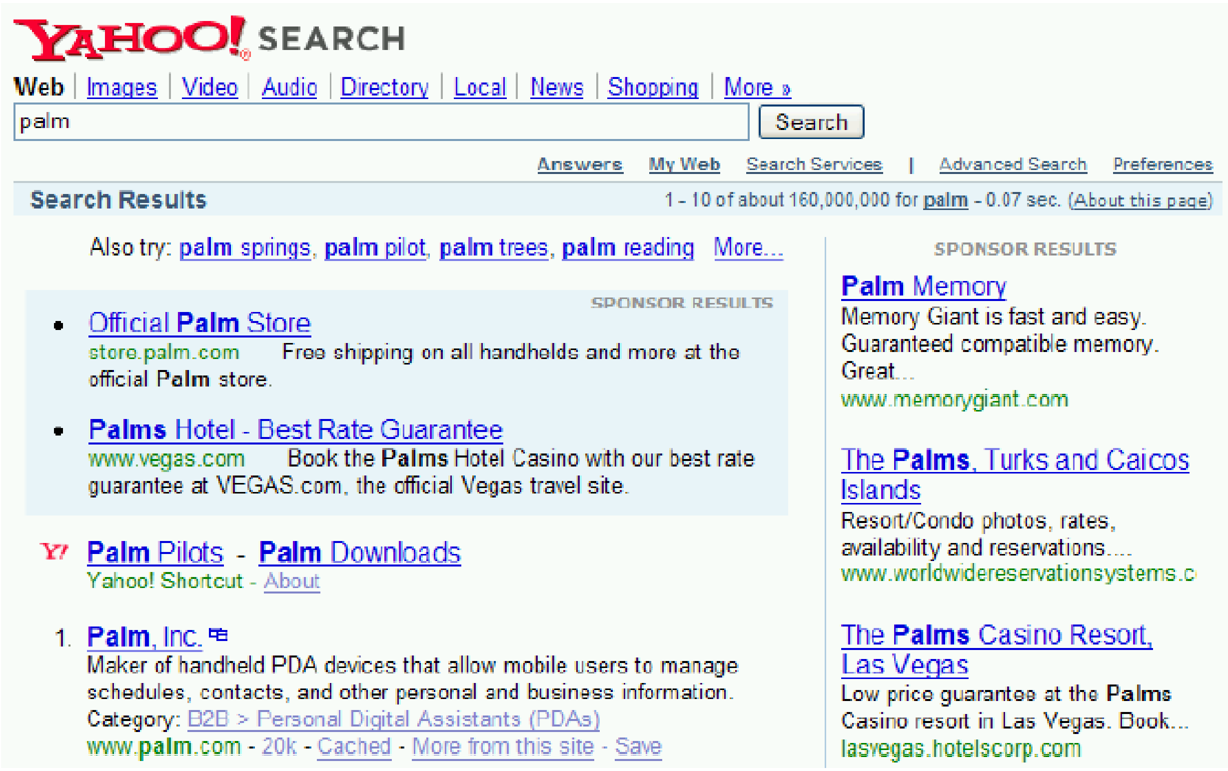
Allow users to visually explore the document space!

Lee and Chen, "Spoken document understanding and organization," *IEEE Signal Processing Magazine* 22 (5), Sept. 2005

- Utilization of local and global analysis techniques to the Web environments
  - How to alleviate the computational burden imposed on the search engine?

Trends and Research Issues (2/3)

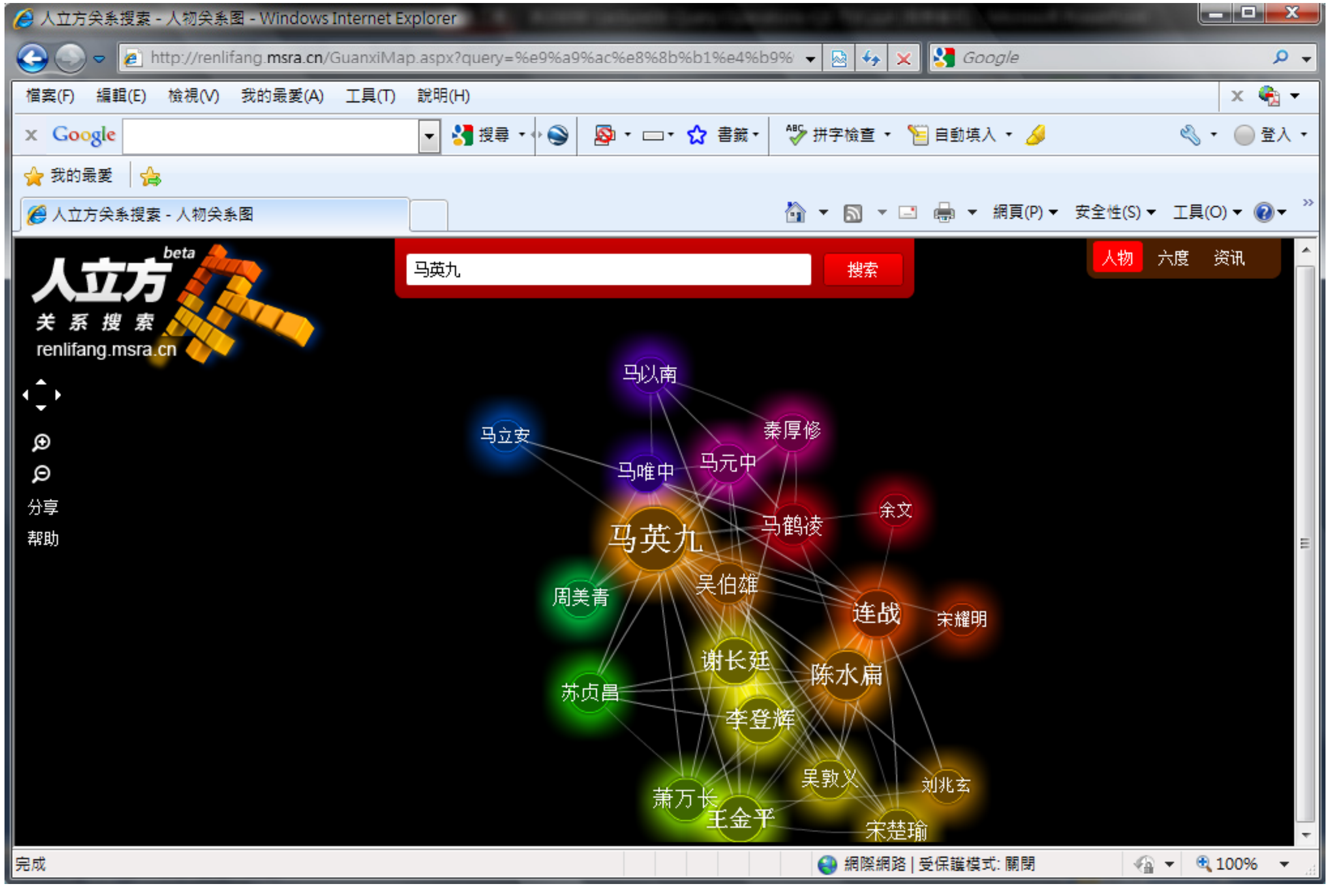
- Yahoo! uses manually built hierarchy of concepts to assist the user with forming the query



► **Figure 9.6** An example of query expansion in the interace of the Yahoo! web search engine in 2006. The expanded query suggestions appear just below the "Search Results" bar.

Trends and Research Issues (3/3)

- Building relationships between named entities
  - Renlifang (人立方) of msra



http://renlifang.msra.cn/