SIGIR 2012 Tutorial
August 12, 2012
Portland Oregon

Beyond Bag-of-Words: Machine Learning for Query-Document Matching in Web Search

Hang Li
Huawei Technologies

Jun Xu
Microsoft

People Who Also Contributed to This Tutorial

Gu XU

Daxin JIANG

Yunhua HU

Jingfang XU

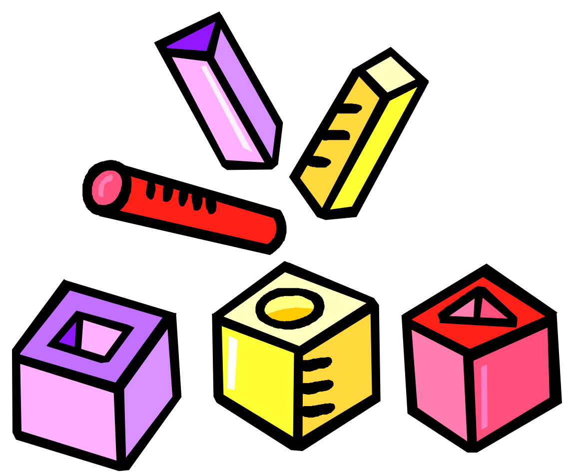
Wei WU

Quan WANG

Outline of Tutorial

1. Learning for Matching between Query and Document (Hang)
2. Matching by Query Reformulation (Hang)
3. Matching with Dependency Model (Jun)
4. Matching with Translation Model (Jun)
5. Matching with Topic Model (Jun)
6. Matching in Latent Space (Hang)
7. Generalization: Learning to Match (Hang)
8. Summary and Open Problems (Hang)

1. Learning for Matching between Query and Document



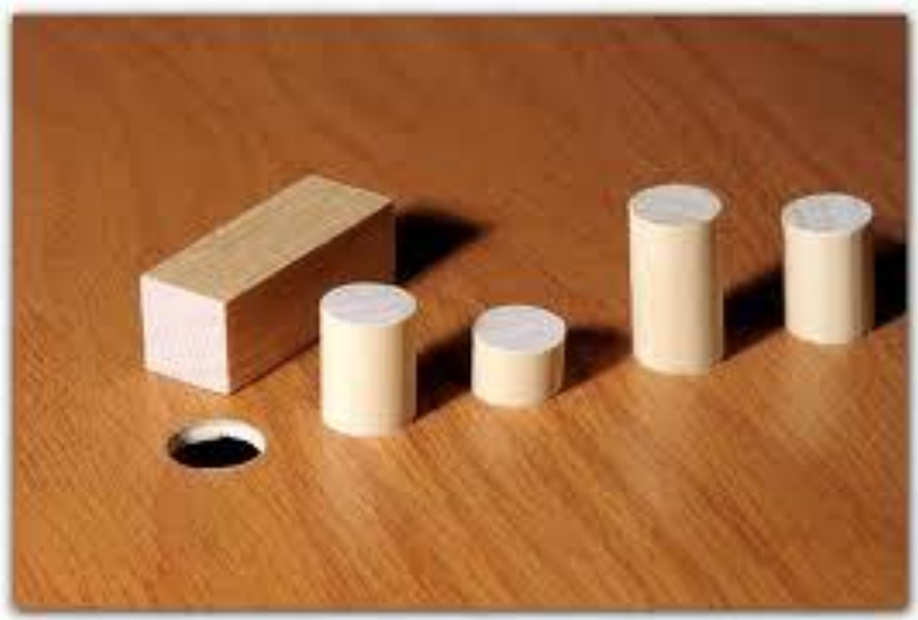
Outline of Section 1

- Query Document Matching in Search
  - Mismatch: Biggest Challenge in Search
  - Matching at Different Levels
  - Matching in Different Ways
- Learning for Matching between Query and Document
- Discussions
  - Relation between Ranking and Matching
  - Previous Work
  - Semantic Matching
  - Long Tail Challenge

A Good Web Search Engine

- Must be good at
  - Relevance
  - Freshness
  - Comprehensiveness
  - User interface
- Relevance is particularly important

Query Document Mismatch is Biggest Challenge in Web Search



Same Search Intent Different Query Representations
Example = “Distance between Sun and Earth”

- "how far" earth sun
- "how far" sun
- "how far" sun earth
- average distance earth sun
- average distance from earth to sun
- average distance from the earth to the sun
- distance between earth & sun
- distance between earth and sun
- distance between earth and the sun
- distance from earth to the sun
- distance from sun to earth
- distance from sun to the earth
- distance from the earth to the sun
- distance from the sun to earth
- distance from the sun to the earth
- distance of earth from sun
- distance between earth sun
- how far away is the sun from earth
- how far away is the sun from the earth
- how far earth from sun
- how far earth is from the sun
- how far from earth is the sun
- how far from earth to sun
- how far from the earth to the sun
- distance between sun and earth

Same Search Intent, Different Query Representations
Example = “Youtube”

• yutube	yuotube	yuo tube
• ytube	youtubr	yu tube
• youtubo	youtuber	youtubecom
• youtube om	youtube music videos	youtube videos
• youtube	youtube com	youtube co
• youtub com	you tube music videos	yout tube
• youtub	you tube com yourtube	your tube
• you tube	you tub	you tube video clips
• you tube videos	www you tube com	wwww youtube com
• www youtube	www youtube com	www youtube co
• yotube	www you tube	www utube com
• ww youtube com	www utube	www u tube
• utube videos	utube com	utube
• u tube com	utub	u tube videos
• u tube	my tube	toutube
• outube	our tube	toutube

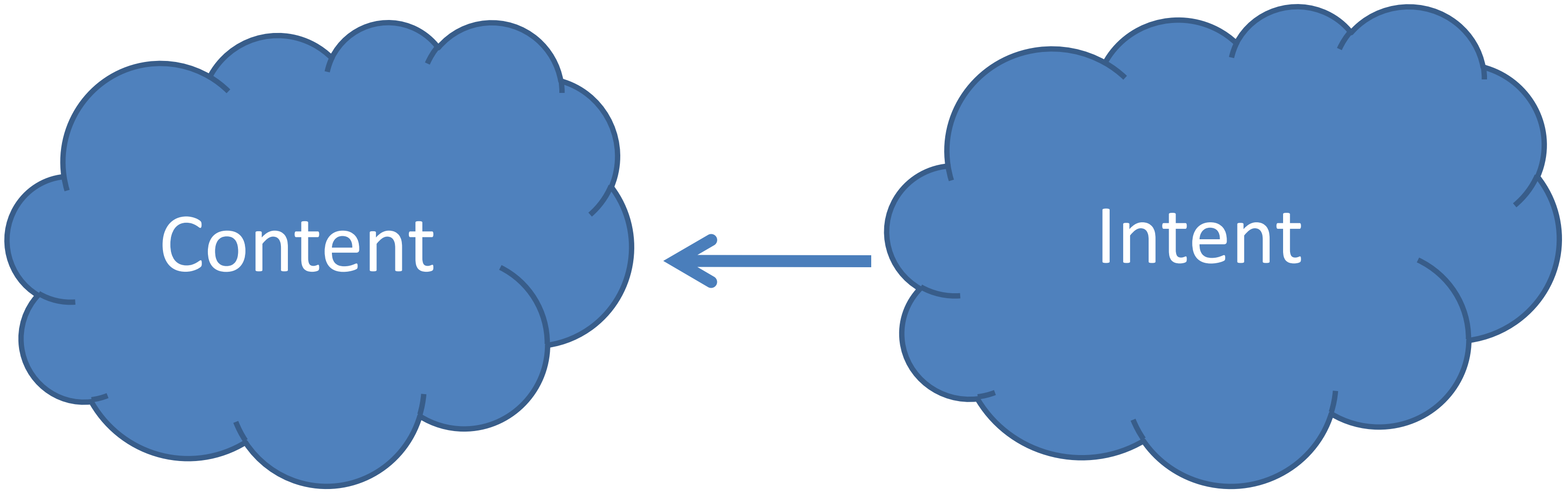
Query Document Mismatch

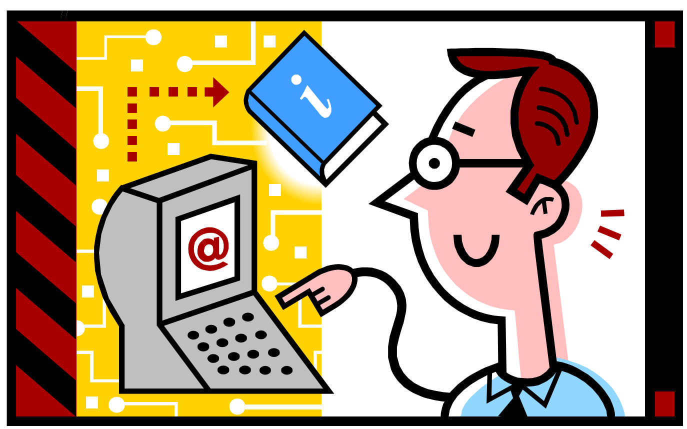
- Same intent can be represented by different queries (representations)
- Search is still mainly based on term level matching
- Query document mismatch occurs, when searcher and author use different representations

Examples of Query Document Mismatch

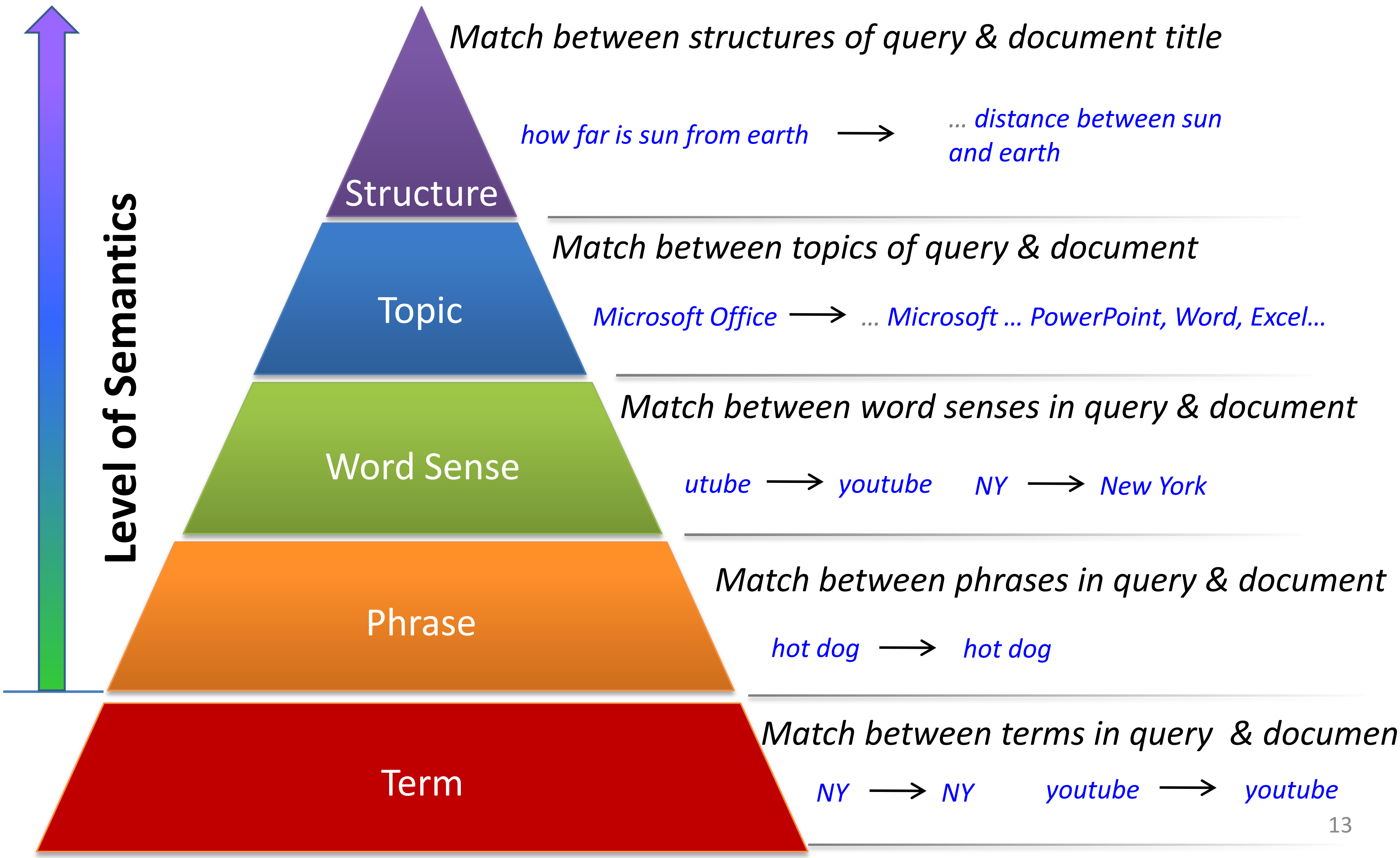
Query	Document	Term Matching	Semantic Matching
seattle best hotel	seattle best hotels	no	yes
pool schedule	swimmingpool schedule	no	yes
natural logarithm transformation	logarithm transformation	partial	yes
china kong	china hong kong	partial	no
why are windows so expensive	why are macs so expensive	partial	no

Matching between Two Worlds:
In Principle, Language Understanding Is Needed





Matching at Different Levels



Query Understanding

Structure Identification

Topic Identification

Similar Query Finding

Phrase Identification

Spelling Error Correction

michael jordan berkele

michael jordan: **main phrase**
Structure

michael jordan berkely: **machine learning**
Topic

michael I. jordan
michael jordan
Sense

[michael jordan] berkeley
Phrase

michael jordan berkeley
Term

Document Understanding

Title Structure Identification

Topic Identification

Key Phrase Identification

Phrase Identification

Homepage of Michael Jordan

Michael Jordan is Professor in the Department of Electrical Engineering

Michael Jordan: ***main phrase in Title***

Structure

Michael Jordan is Professor in the Department of Electrical Engineering: ***machine learning***

Topic

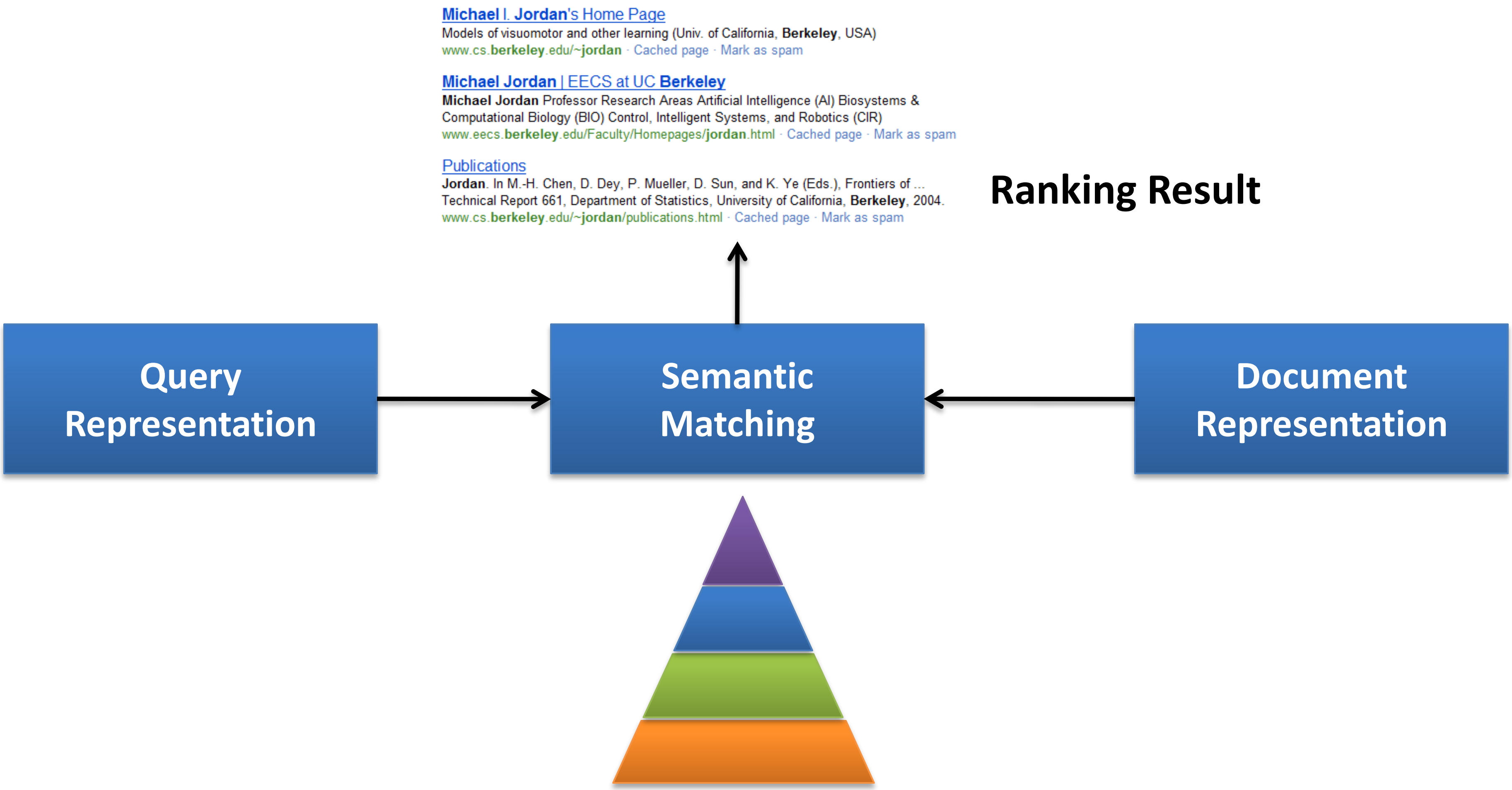
[Michael Jordan], [Professor]
[Electrical Engineering]: ***keyphrase***

Phrase

[Michael Jordan] is [Professor] in the [Department] of [Electrical Engineering]

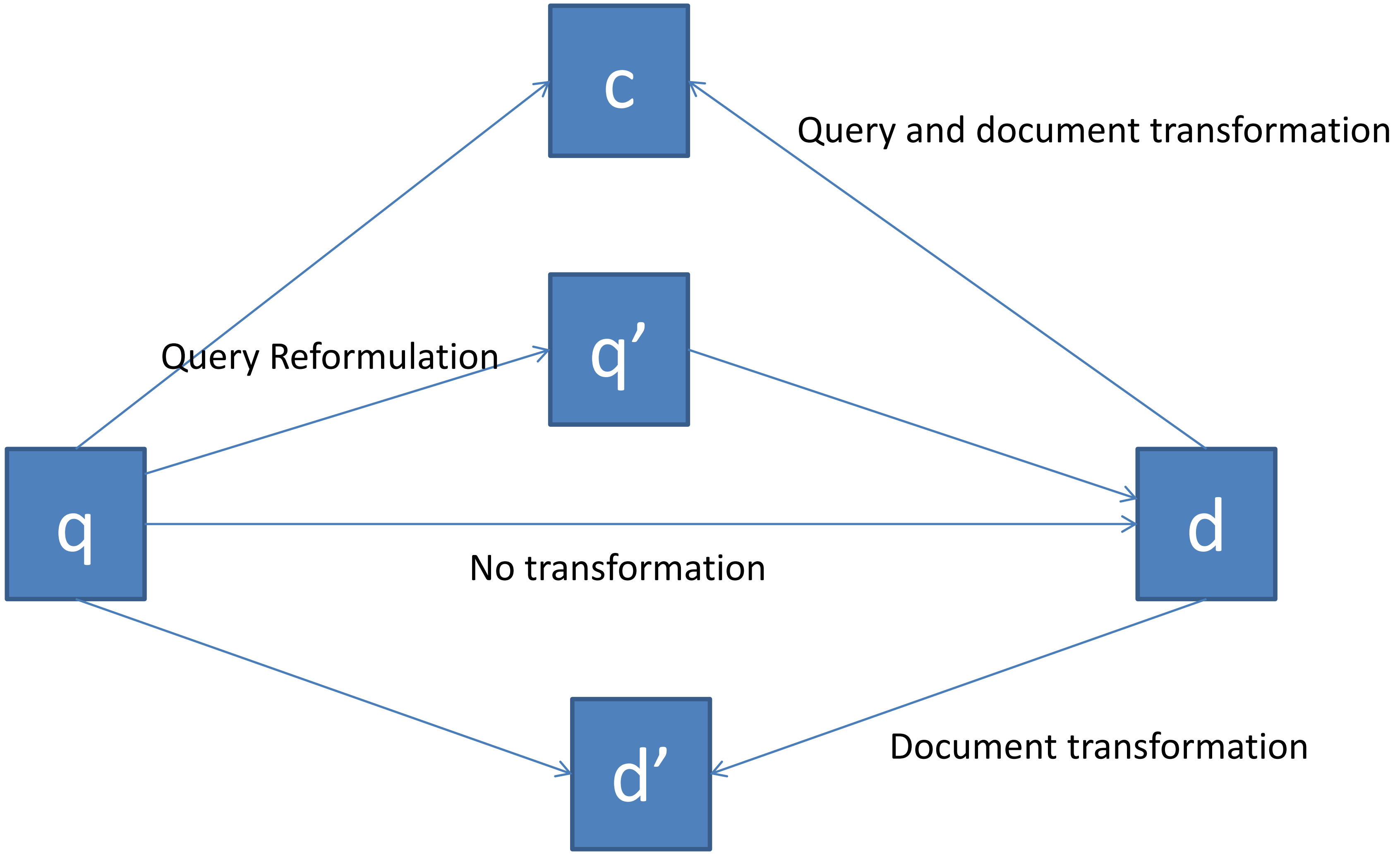
Phrase

Online Matching

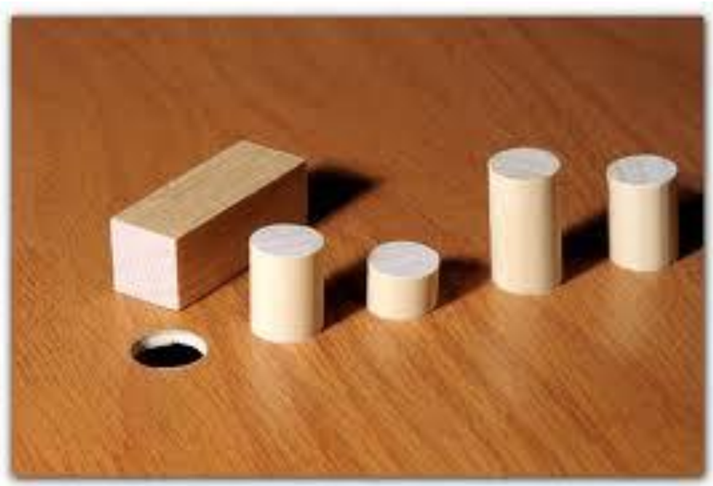


Matching can be conducted at different levels

Matching in Different Ways



Machine Learning for Query Document Matching in Web Search



Learning for Matching between Query and Document

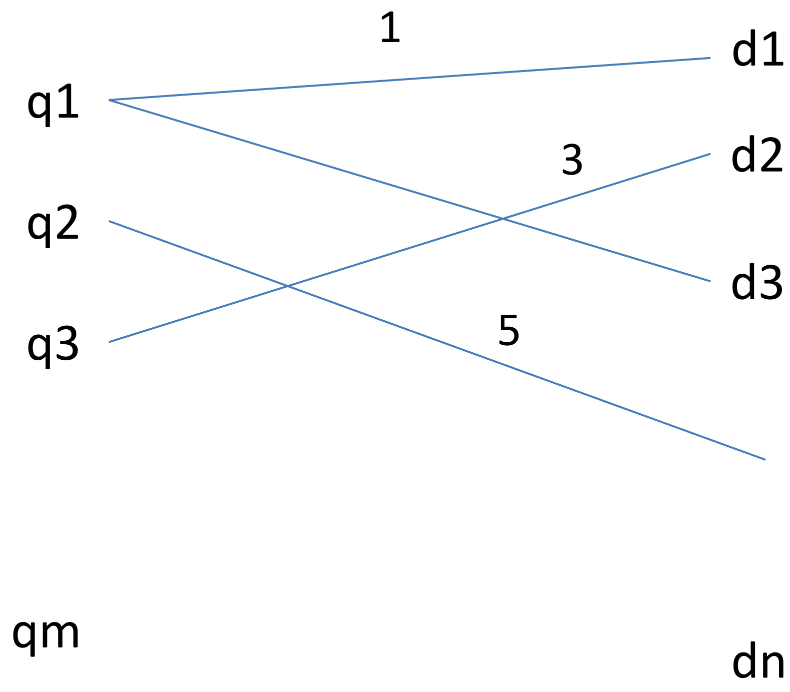
- Learning matching function

  $f_M(q,d)$ or $p_M(r \mid q,d)$

- Using training data $(q_1, d_1, r_1), \cdots, (q_N, d_N, r_N)$
- $q_1, q_2, \cdots, q_N$ and $d_1, d_2, \cdots, d_N$ can be id's or feature vectors
- $r_1, r_2, \cdots, r_N$ can be binary or numerical values
- *Using relations in data and/or prior knowledge*

Matching Problem: Instance Matching

Graph View

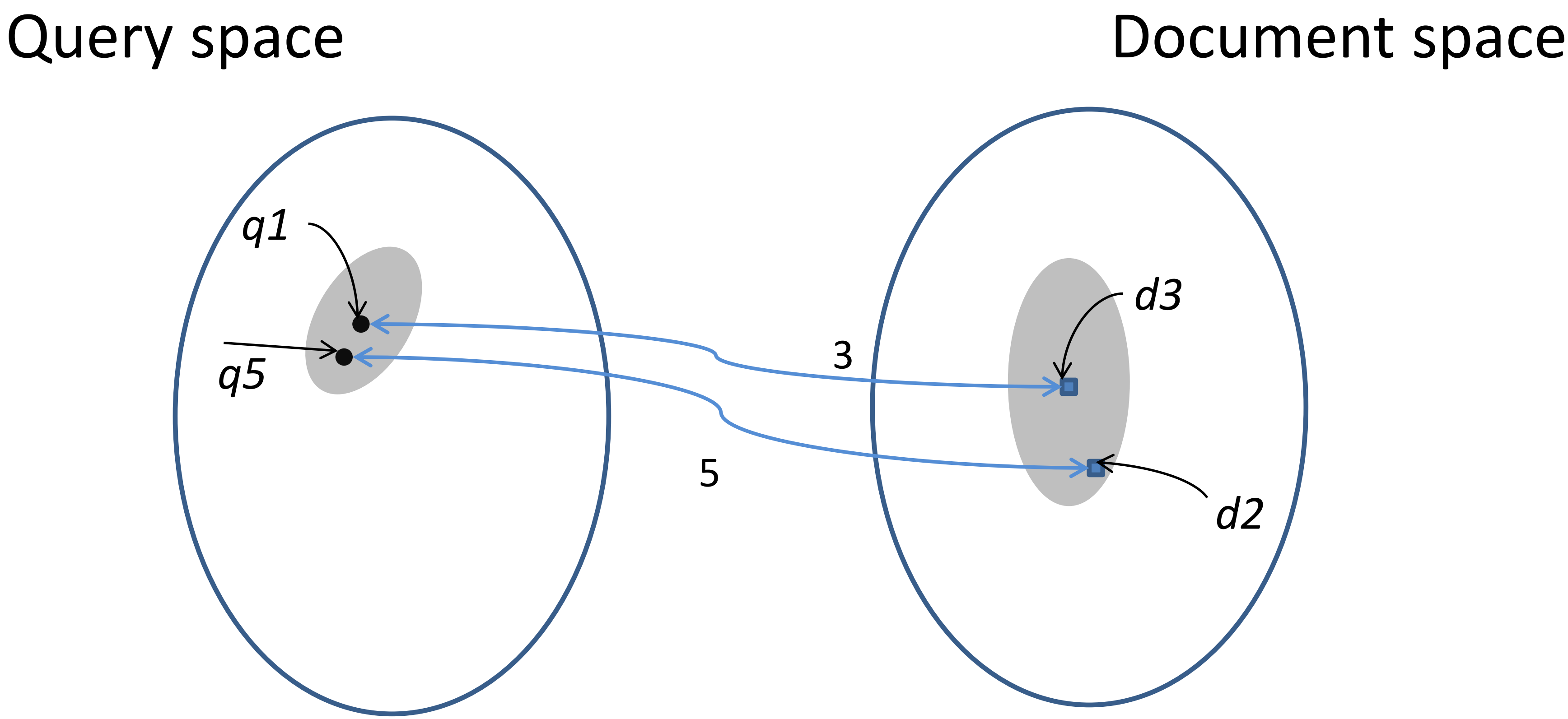


Matching Problem: Instance Matching

Matrix View

	d1	d2	d3			dn
q1			1			
q2						1
q3				4		
		1			5	
qm						

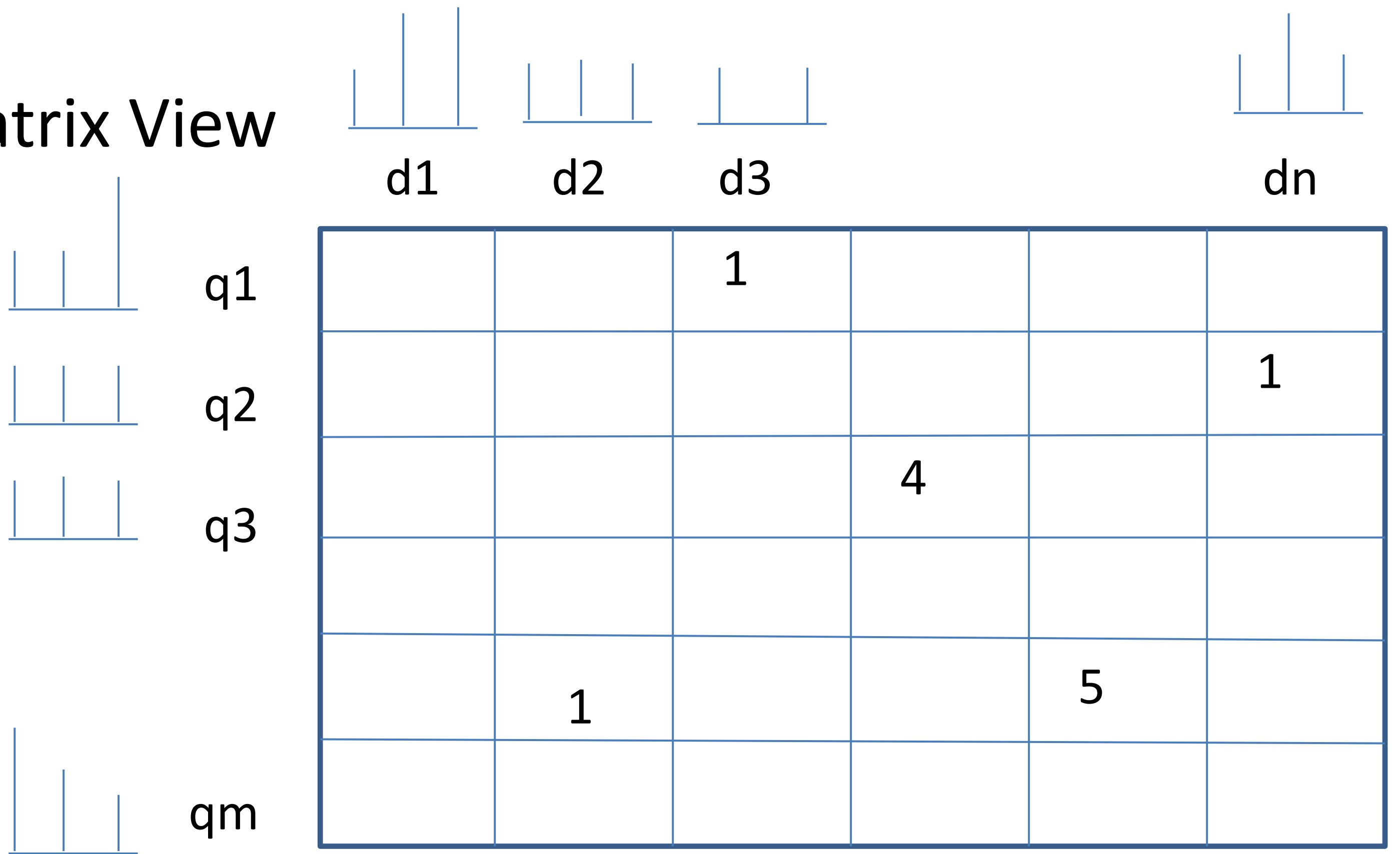
Matching Problem: Content Matching



Space View

Matching Problem: Content Matching

Matrix View



Challenges in Machine Learning for Matching

- How to leverage relations in data and prior knowledge
- Scale is very large

Relation between Matching and Ranking

- In traditional IR:
  - Ranking = matching

$$f(q,d) = f_{BM25}(q,d) \quad \text{or} \quad f(q,d) = P_{LMIR}(d \mid q)$$

- Web search:
  - Ranking and matching become separated
  - Learning to rank becomes state-of-the-art

$$f(q,d) = f_{BM25}(q,d) + g_{PageRank}(d) + \cdots$$

  - Matching = feature learning for ranking

Matching vs Ranking

In search, first matching and then ranking

	Matching	Ranking
Prediction	Matching degree between query and document	Ranking list of documents
Model	f(q, d)	f(q,d1), f(q,d2), … f(q,dn)
Challenge	Mismatch	Correct ranking on top

Matching Functions as Features in Learning to Rank

- Term level matching: $f_{BM25}(q,d)$ $f_{n-BM25}(q,d)$
- Phrase level matching: $f_P(q,d)$
- Sense level matching: $f_S(q,d)$
- Topic level matching: $f_T(q,d)$
- Structure level matching: $f_C(q,d)$
- Term level matching (spelling, stemming): $q' \rightarrow q$

Linear Combinations of Matching Functions

- Query Reformulation

$$f(q,d) = f_{BM25}(q,d) + \sum_i k_Q(q,q_i) k_D(d,d_i) f(q_i,d_i)$$

- Topic Model

$$f(q,d) = f_{LMIR}(q,d) + \sum_k u(q,k) v(k,d)$$

Previous Work

- Studied in long history of IR
- Query expansion, pseudo relevance feedback
- Latent Semantic Indexing, Probabilistic Latent Semantic Indexing
- … …

New Trends in Recent Work

- Employing more machine learning (supervised and unsupervised)
- Large scale
- Use of log data

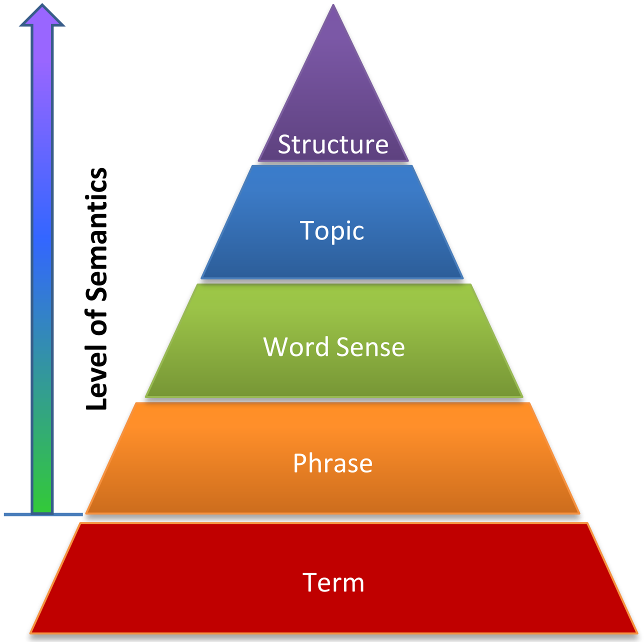
- This tutorial focuses on recent work!

Previous Work v.s. Recent Work

	Previous	Recent
Scale	Small	Large
Methodologies	Unsupervised learning	Both supervised learning and unsupervised learning
Data	No use of log data	Use of log data

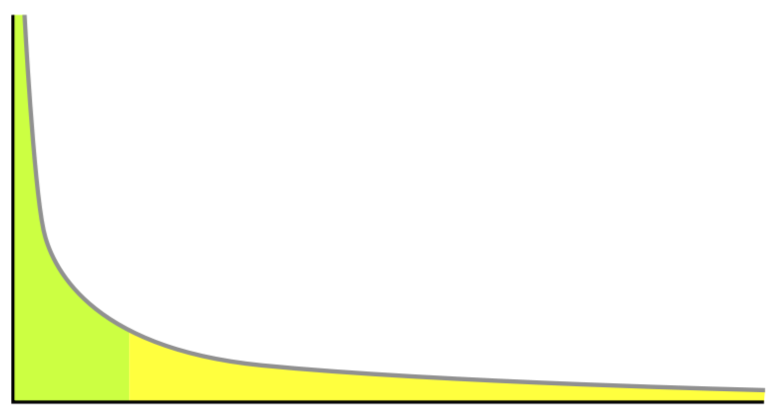
Semantic Matching

- Matching based on "semantics", i.e., topics, sense, structure
- Beyond traditional term matching
- Ultimate goal: language understanding



Long Tail Challenge

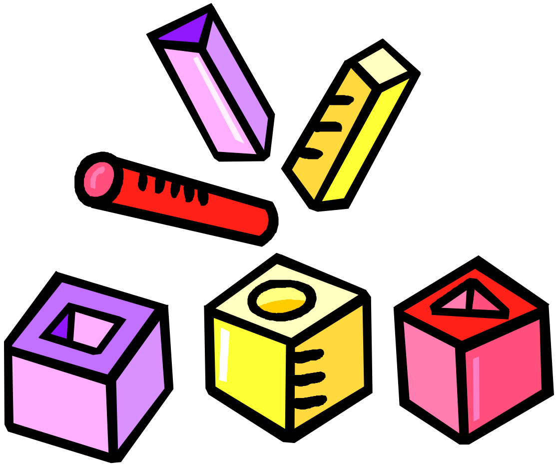
- Head pages have rich anchor texts and click data
- Tail queries and pages suffer more from mismatch
- Problem of propagating information and knowledge from head to tail



Approaches to Learning for Matching Between Query and Document

- Matching by Query Reformulation
- Matching with Dependency Model
- Matching with Translation Model
- Matching with Topic Model
- Matching in Latent Space

2. Matching by Query Reformulation



Outline of Section 2

- Query Reformulation
- Problems in Query Reformulation
  - Query Reformulation
  - Blending
  - Similar Query Mining
- Methods of Query Reformulation
- Methods of Blending
- Methods of Similar Query Mining
- QRU-1 Dataset

Query Reformulation Is Also Called

- Query Transformation
- Query Rewriting
- Query Refinement
- Query Alteration

- Terminology regarding to Query Representation and Understanding (Croft et al., '10)

Query Transformation

- Our focus is on how queries can be *transformed* to equivalent, potentially better, queries
  - Queries into paraphrases or “translations”
  - Long queries into shorter queries
  - Short queries into longer queries
  - Queries in one domain to queries in other domains
  - Unstructured queries into structured queries

From Bruce Croft, ECIR 2009

Types of Query Reformulation

- Spelling Error Correction
  - 10-15% queries contain spelling errors
  - E.g., “mlss singapore” → “miss singapore” ×
    mlss=machine learning summer school
- Merging
  - E.g., “face book” → “facebook”
- Splitting
  - E.g., “dataset” → “data set”
- Query Segmentation
  - E.g., “new york time square” → “(new york) (time square)”

Types of Query Reformulation (2)

- Stemming
  - E.g, “seattle best hotel” → “seattle best hotels”
- Synonym
  - E.g, “ny times” → “new york times”
- Paraphrasing
  - E.g., “how far is sun from earth” → “distance between sun and earth”
- Query Expansion
  - E.g., “www” → “www conference”
- Query Deduction
  - E.g., “natural logarithm transformation” → “logarithm transformation”

Problems in Query Reformulation

- Query Reformulation
- Blending
- Similar Query Mining

Query Reformulation Problem

- Task
  - Rewrite original query to multiple similar queries
- Challenges
  - Topic drift
- Current Situation
  - Mainly limited to auto correction of spelling errors in practice

Query Reformulation is Difficult

- Depending on the contents of both query and document
- Except
  - Spelling error correction
  - *Definite* splitting and merging, e.g., “facebook”
  - *Definite* segmentation, e.g., “hot dog”, “united states”

Methods of Query Reformulation

- Generative Approach:
  - Source Channel Model (Brill & More, ’00)
  - Source Channel (Cucerzan & Brill, ’04)
  - Source Model (Duan & Hsu, ’10)
- Discriminative Approach:
  - MaxEnt (Li et al., ’06)
  - Log Linear Model (Okazaki et al., ’08)
  - Log Linear Model (Wang et al., ’11)
  - Conditional Random Field (Guo et al., ’08)

Source Channel Model (Brill & Moore, 2000; Duan & Hsu, 2011)

- Source Channel Model

$$\hat{c} = \arg\max_{c} P(c \mid q)$$

$$= \arg\max_{c} P(q \mid c) P(c)$$

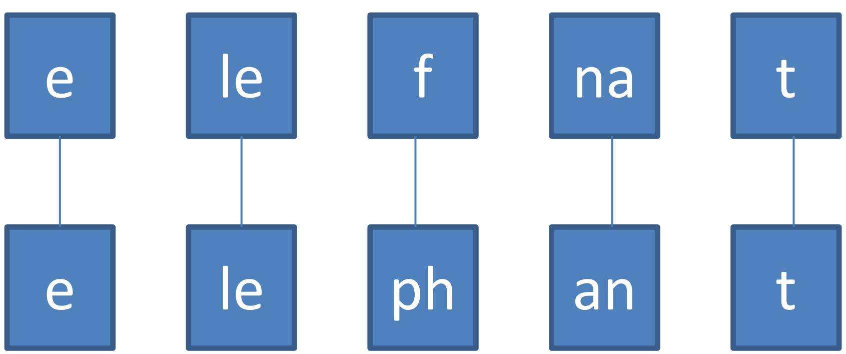
- Source Model (Language Model) $P(c)$
- Channel Model (Transformation Model) $P(q \mid c)$

Transformation Model

- Model

$$P(q \mid c) = \sum_{s \in S(c \to q)} \prod_{i=1}^{l^s} P(t_0 \mid t_{i-M+1} \cdots t_{i-1})$$

- Sequence of Transfemes



Learning and Prediction

- Parameter Estimation
  - EM Algorithm
  - Pruning
  - Smoothing
- Search
  - Trie: encoding dictionary
  - A* Algorithm

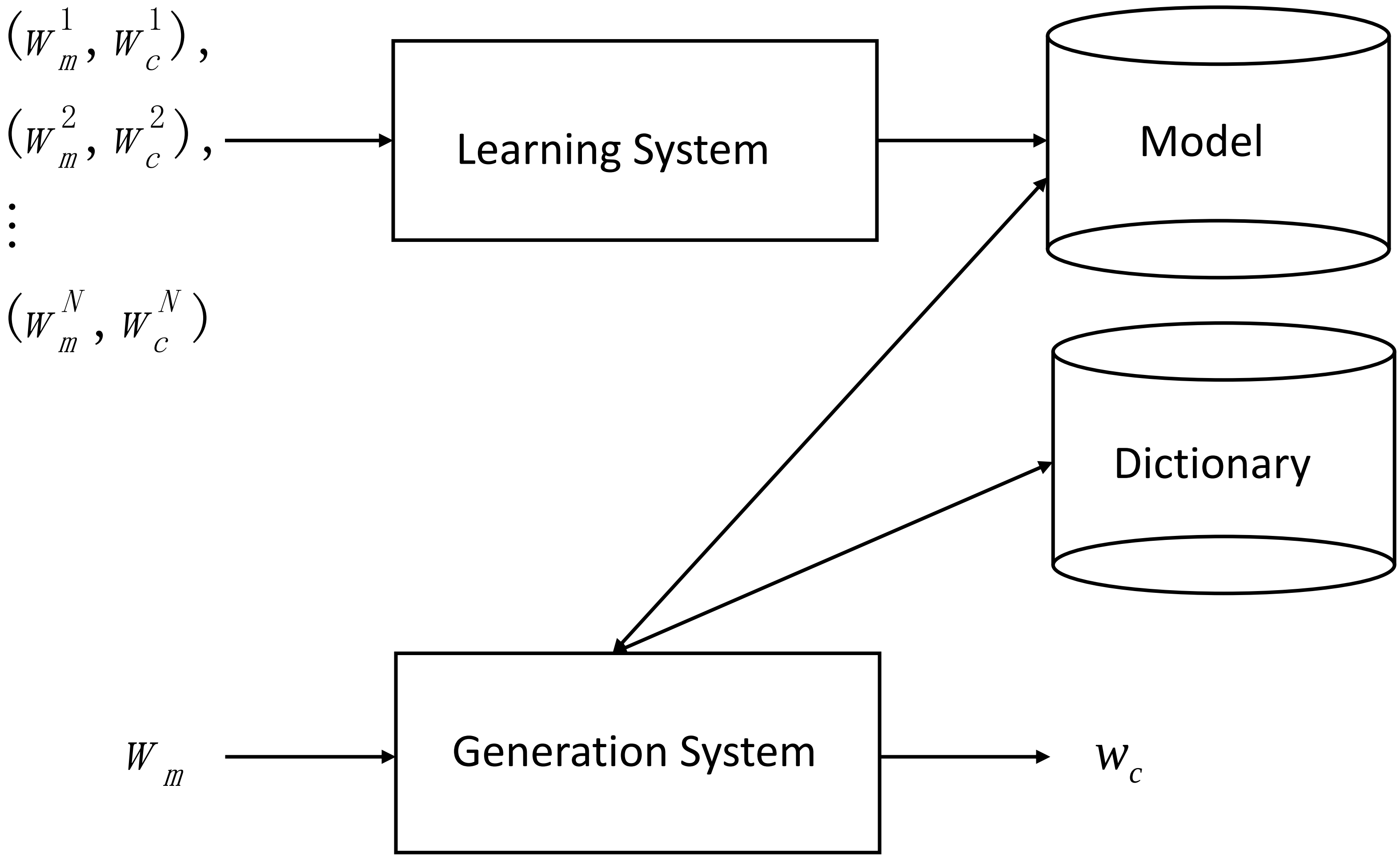
Log Linear Model (Wang et al, 2011)

- Query reformulation $q_m \rightarrow q_c$
- Transformation rules $R(q_m, q_c)$
- Learning

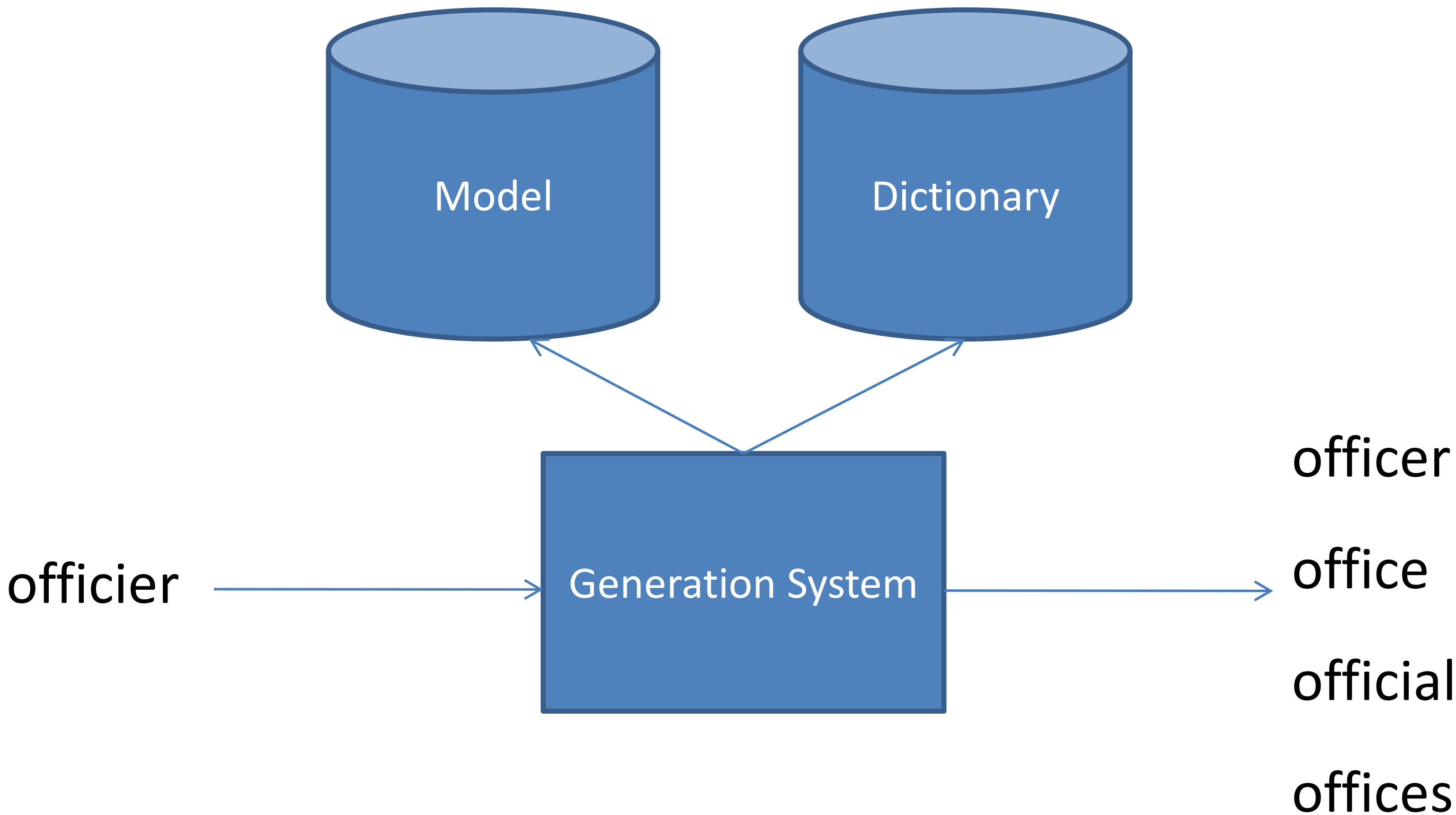
  $P(q_c, R(q_m, q_c) \mid q_m)$
- Prediction

  $\max_{q_c, R} P(q_c, R(q_m, q_c) \mid q_m)$
- Can be used at both word level and query level
- Model = log linear model
- Both accurate and efficient

Learning and Prediction



Example: Spelling Error Correction



Learning

Training Data → **Rule Extraction** → **Model Learning** → **Model**

(w_m^1, w_c^1)

(w_m^2, w_c^2)

(w_m^3, w_c^3)

$\cdots$

$\alpha_1 \rightarrow \beta_1$

$\alpha_2 \rightarrow \beta_2$

$\alpha_3 \rightarrow \beta_3$

$\cdots$

rule

$P(w_c, R(w_m, w_c) \mid w_m)$

log linear model

$\alpha_1 \rightarrow \beta_1, \lambda_1$

$\alpha_2 \rightarrow \beta_2, \lambda_2$

$\alpha_3 \rightarrow \beta_3, \lambda_3$

$\cdots$

weight

Rule Extraction

- Edit-distance based alignment:

Misspelled:	^	*n*	*i*	*c*	*o*	*s*	*o*	*o*	*f*	*t*	*$*
	↓	↓	↓	↓	↘	↘		↓	↓	↓	↓
Correct:	^	*m*	*i*	*c*	*r*	*o*	*s*	*o*	*f*	*t*	*$*

- Basic substitution rules:

$$n \rightarrow m, \phi \rightarrow r$$

- Contextual substitution rules

$$\hat{}n \rightarrow \hat{}m, ni \rightarrow mi, \hat{}ni \rightarrow \hat{}mi, c \rightarrow cr, \ldots$$

Log Linear Model

- Model

$$P(w_c, R(w_m, w_c) \mid w_m) = \frac{\exp(\sum_{r \in R(w_m, w_c)} \lambda_r)}{\sum_{(w_c', R(w_m, w_c')) \in Z(w_m)} \exp(\sum_{o \in R(w_m, w_c')} \lambda_o)}$$

Weight of rule

Set of rules rewrite w_m to w_c

All pairs of word w'_c and rule set $R(w_m, w'_c)$

$$\forall \lambda_r \le 0$$

Non-positive constraint, to improve efficiency in retrieval, Natural assumption

- Candidate Generation

$$rank(w_c \mid w_m) = \max_{R(w_m, w_c)} (\sum_{r \in R(w_m, w_c)} \lambda_r)$$

Model Parameter Estimation

- Objective function

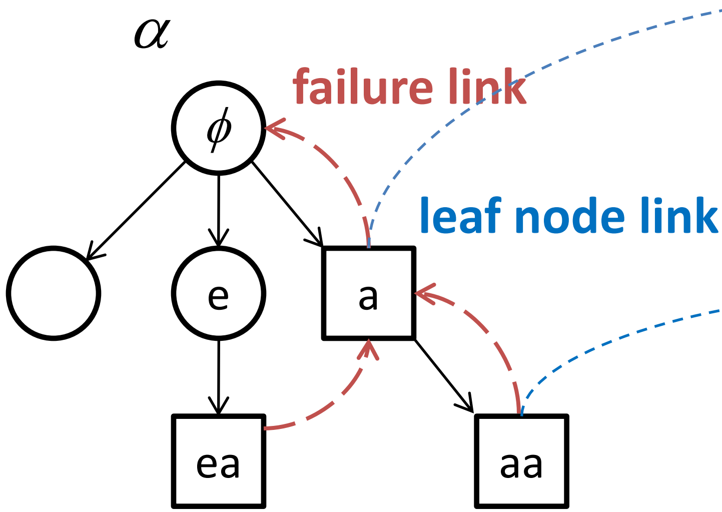
$$\lambda^* = \arg\max_{\lambda} \sum_i \log \underbrace{\sum_{R(w_m^i, w_c^i)}}_{\text{Take max over transformations}} P(w_c^i, R(w_m^i, w_c^i) \mid w_m^i)$$

Take max over transformations

- Algorithm
  - Constrained Quasi Newton Method (BFGS)

Matching with Rules Using Aho Corasick Tree

β are stored in an associated list



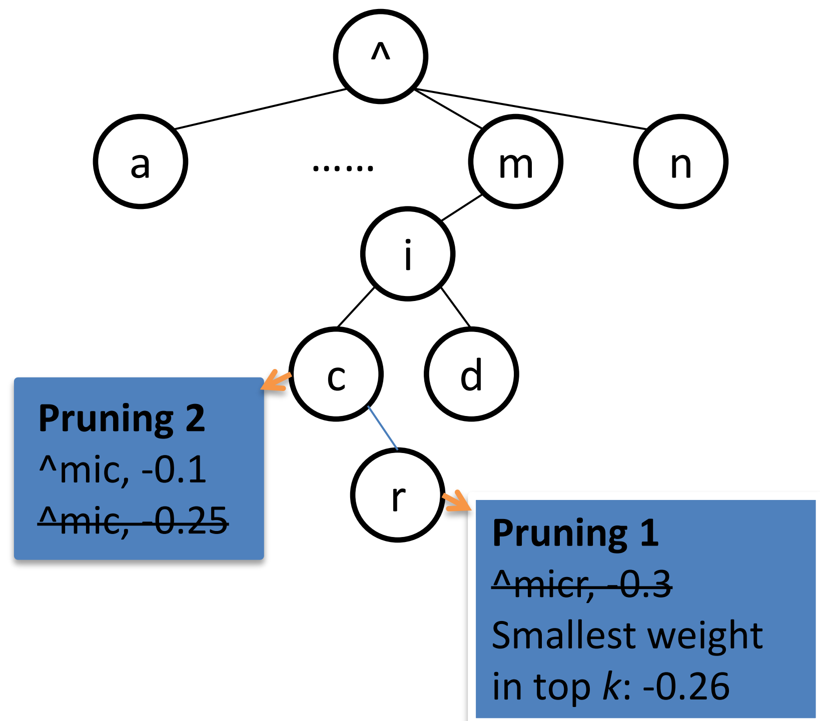
β

e	0.0	$a \rightarrow e$
s	-0.3	$a \rightarrow s$
……	……	
NULL	—	
a	-0.1	$aa \rightarrow a$
……	……	
NULL	—	
……	……	

Index all the αs in the rules on the AC tree

Matching with Dictionary Using Trie Tree

- Traverse trie tree
  - Match the next position of w_m
  - Apply a rule at the current position of w_m
- Two pruning strategies
  - If the sum of weights is smaller than the smallest weight in the top k list, prune the branch
  - two search branches merge, prune the smaller branch



Conditional Random Field (Guo et al, 2008)

- Sequential Prediction
- Learning

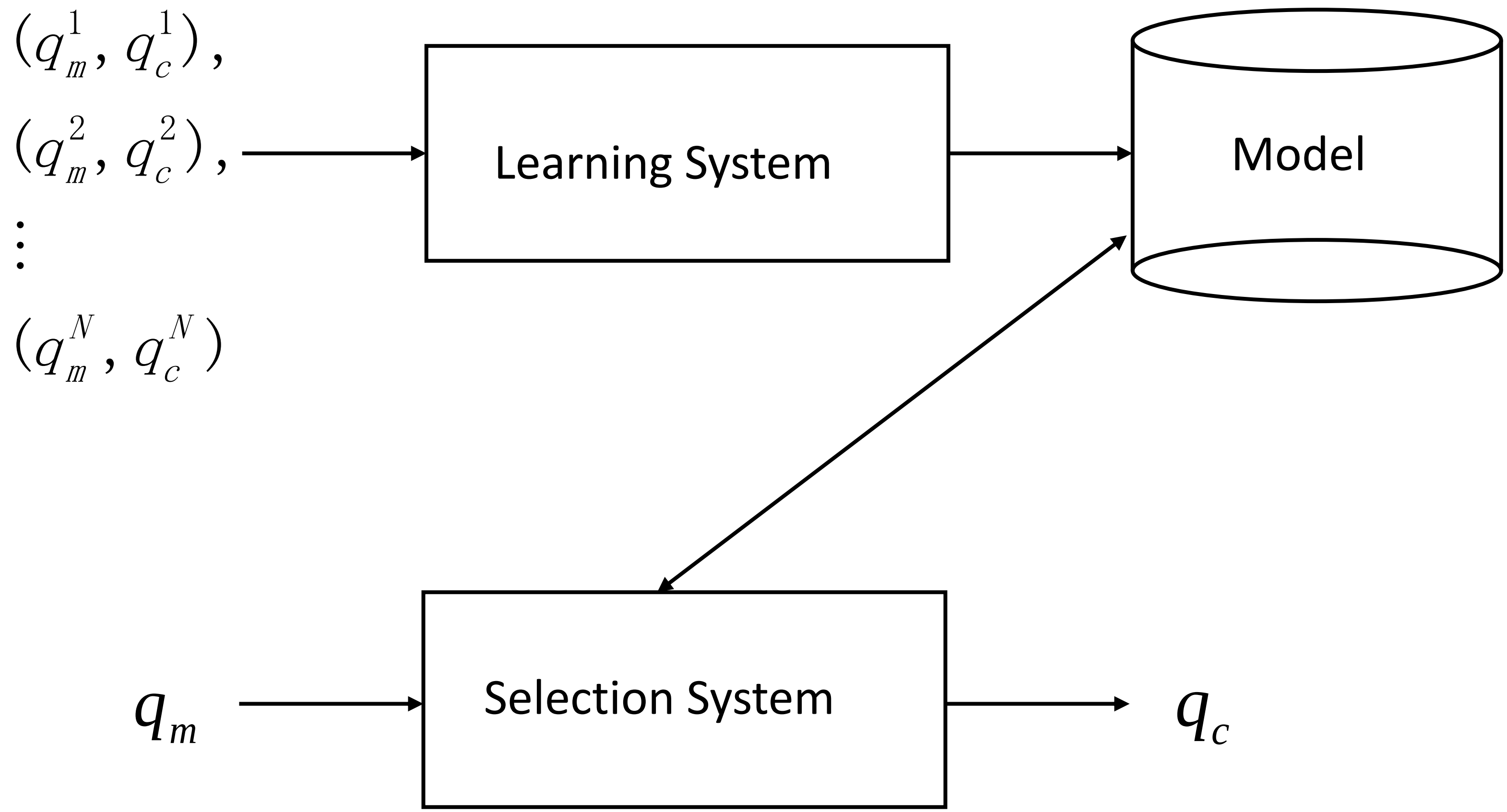
$$P(q_c, o \mid q_m)$$

- Prediction

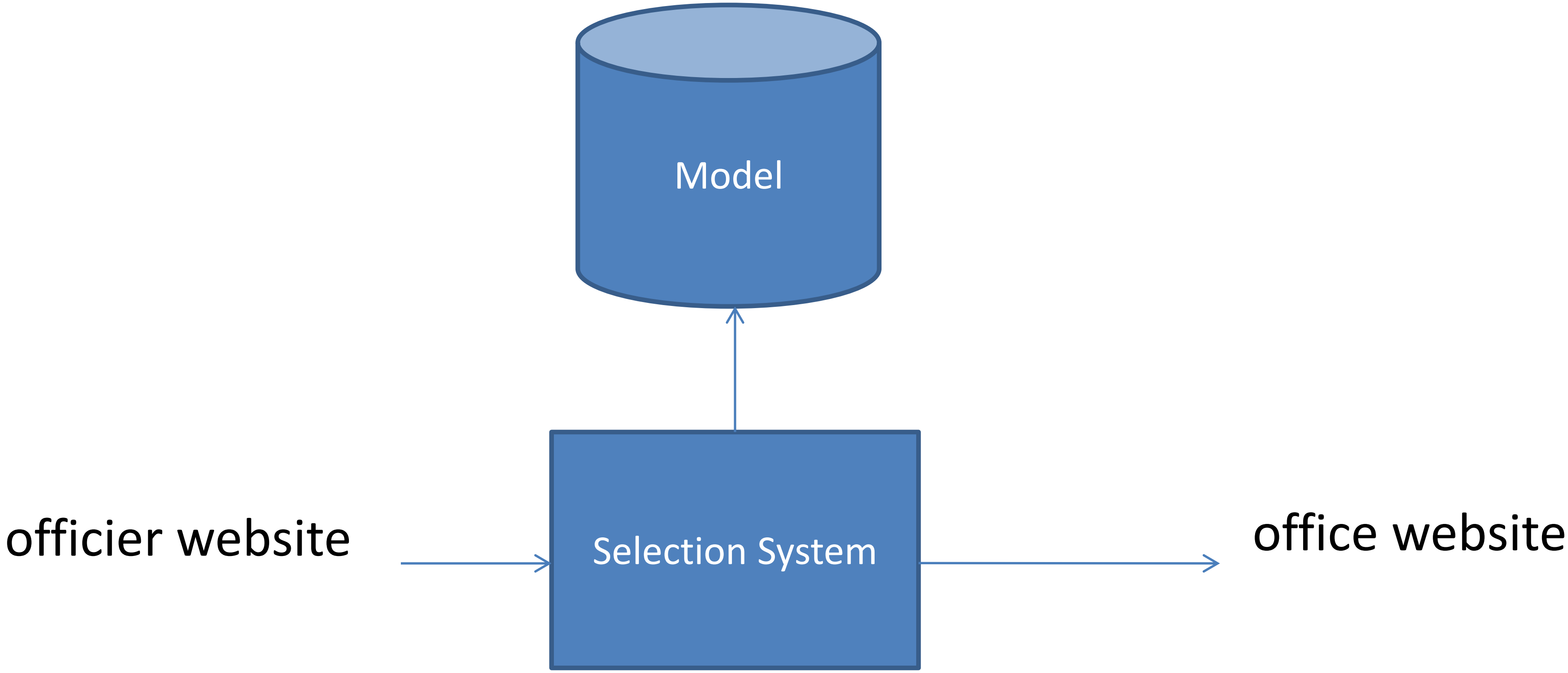
$$\max_{q_c, o} P(q_c, o \mid q_m)$$

- Can be used at both word level and query level
- Model = conditional random field
- A general word of query reformulation

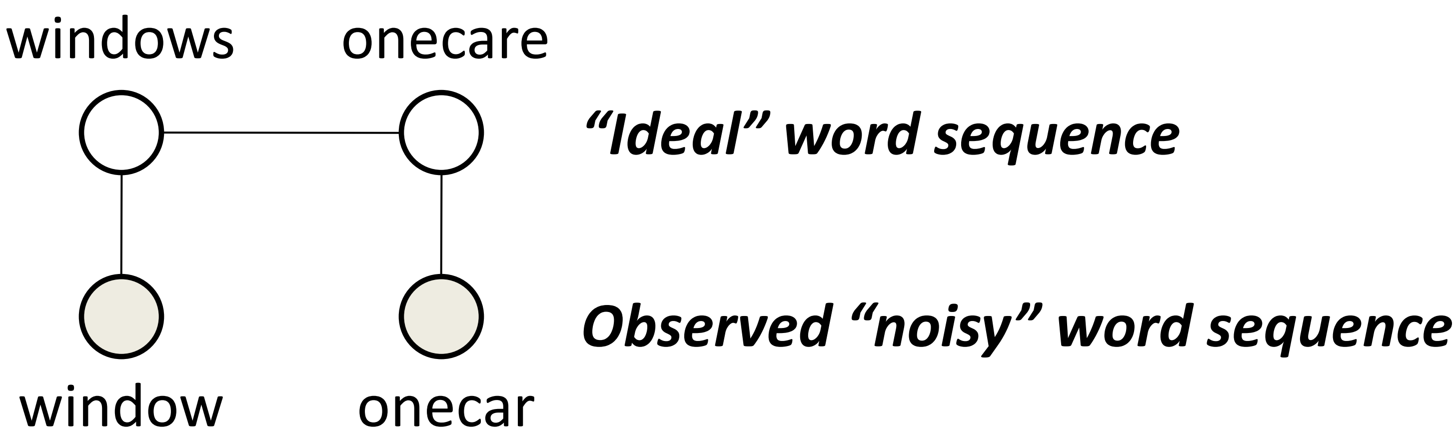
Learning and Prediction



Example: Spelling Error Correction



Candidate Selection Problem



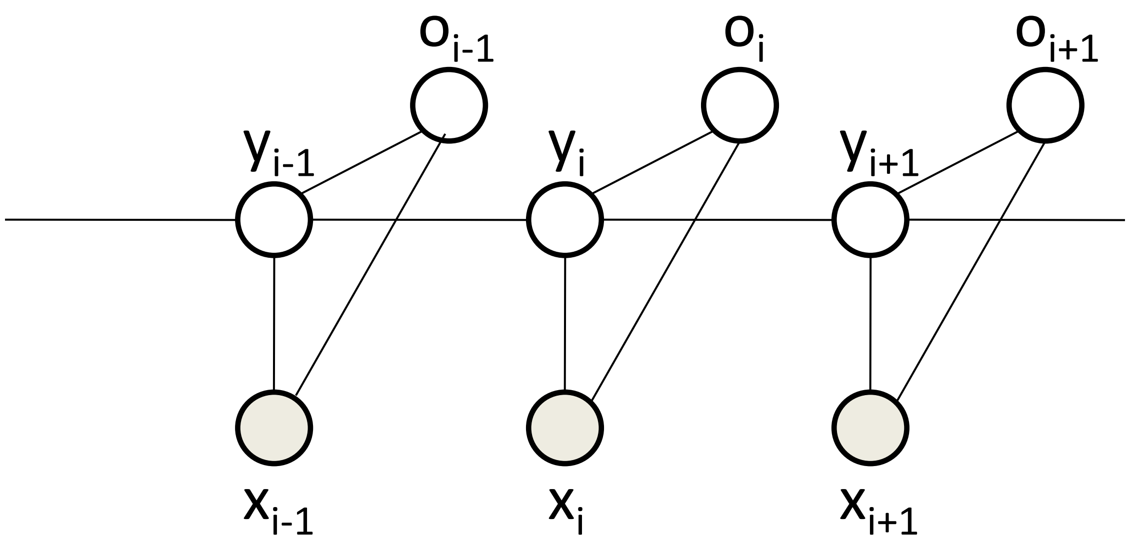
$$y^* = \arg\max_y \Pr(y|x)$$

"ideal" query word sequence

original query word sequence

Conditional Random Field

Introducing Refinement Operations



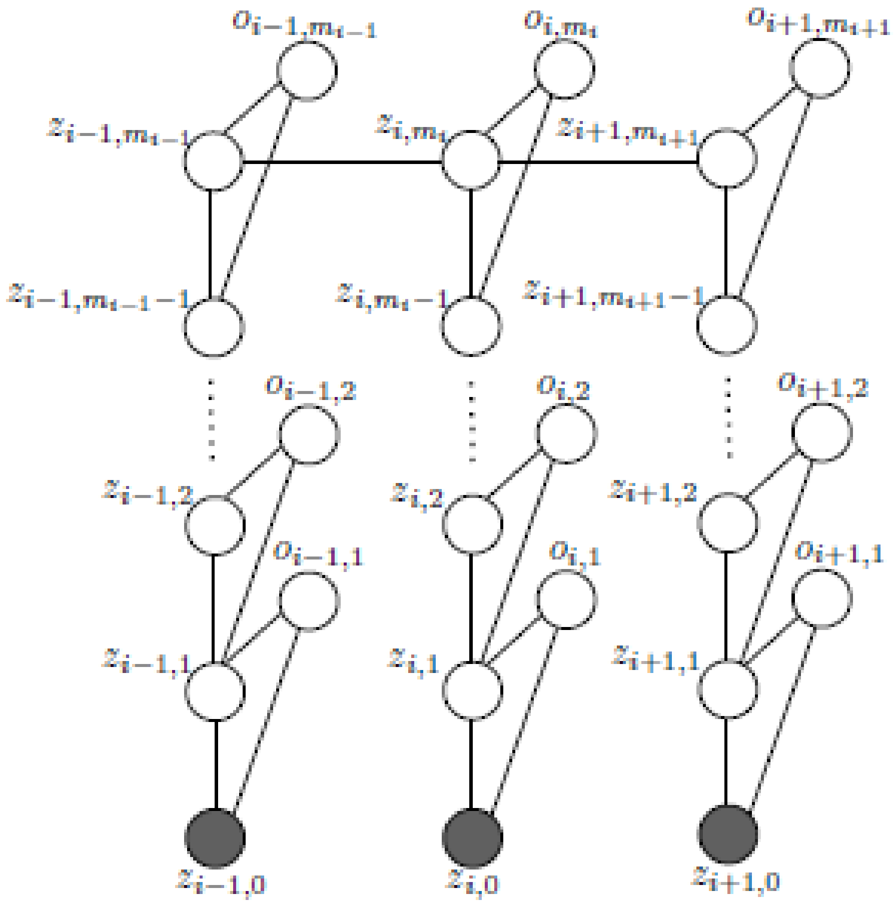
$$\Pr(y, o|x) = \frac{1}{Z(x)} \prod_{i=1}^{n} \phi(y_{i-1}, y_i)\phi(y_i, o_i, x)$$

Operations

Spelling: insertion, deletion, substitution, transposition, …

Word Stemming: +s/-s, +es/-es, +ed/-ed, +ing/-ing, …

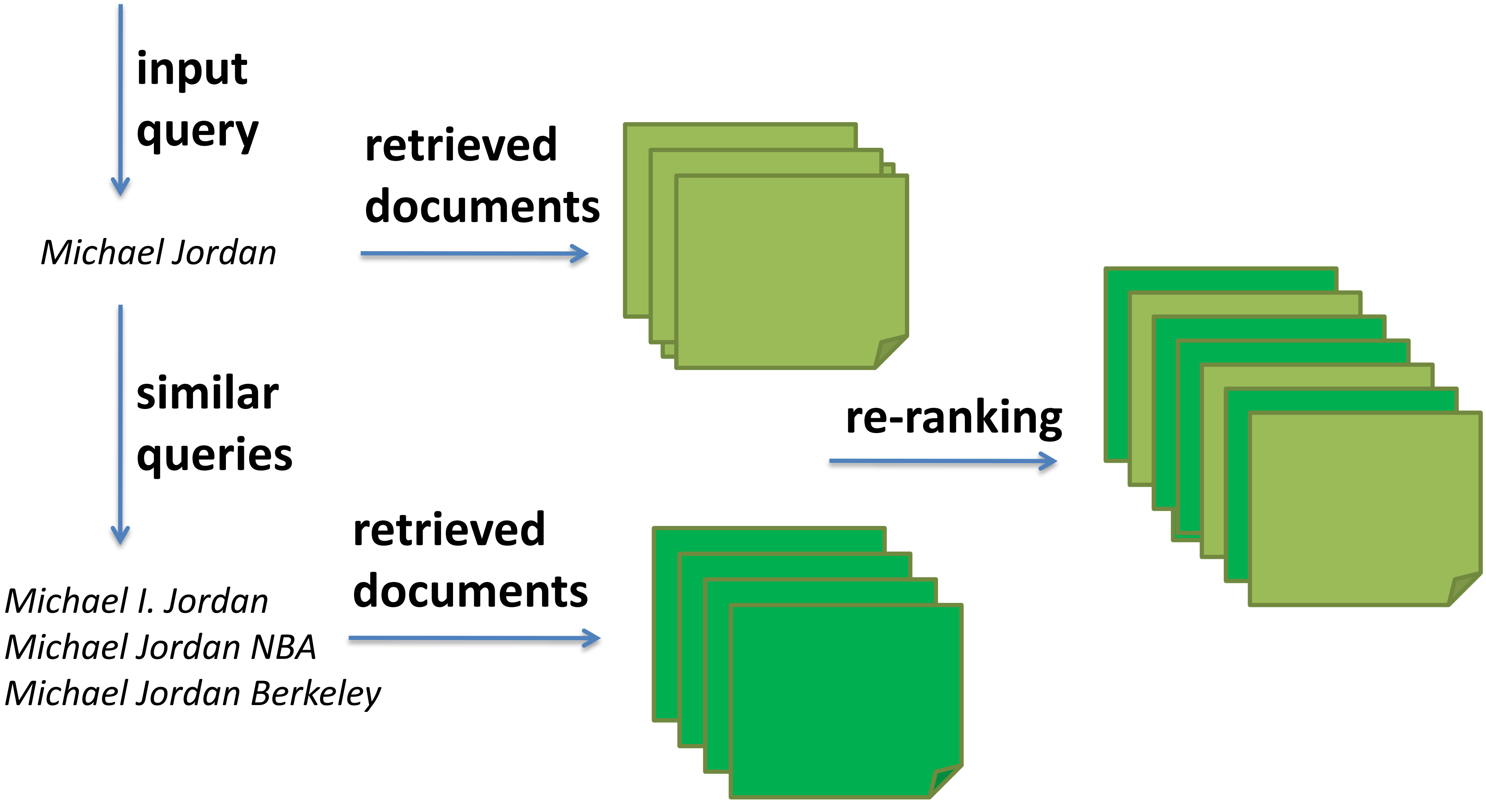
Extended Conditional Random Fields



Blending Problem

- Steps
  - Rewrite original query to multiple similar queries
  - Retrieve with multiple queries
  - Blend results from multiple queries
- Challenges
  - System to sustain searches with multiple queries
  - Blending model: matching scores are not comparable across queries

Blending


input
query
Michael Jordan
retrieved
documents
similar
queries
Michael I. Jordan
Michael Jordan NBA
Michael Jordan Berkeley
retrieved
documents
re-ranking

Methods of Blending

- Linear Combination (Xue et al., ’08)
- Learning to Rank (Sheldon et al., ’11)
- Kernel Methods (Wei et al., ’11)

Linear Combination

- Matching model

$$f(q,d) = f_{LMIR}(q,d) + \sum_i k_Q(q,q_i) f(q_i,d)$$

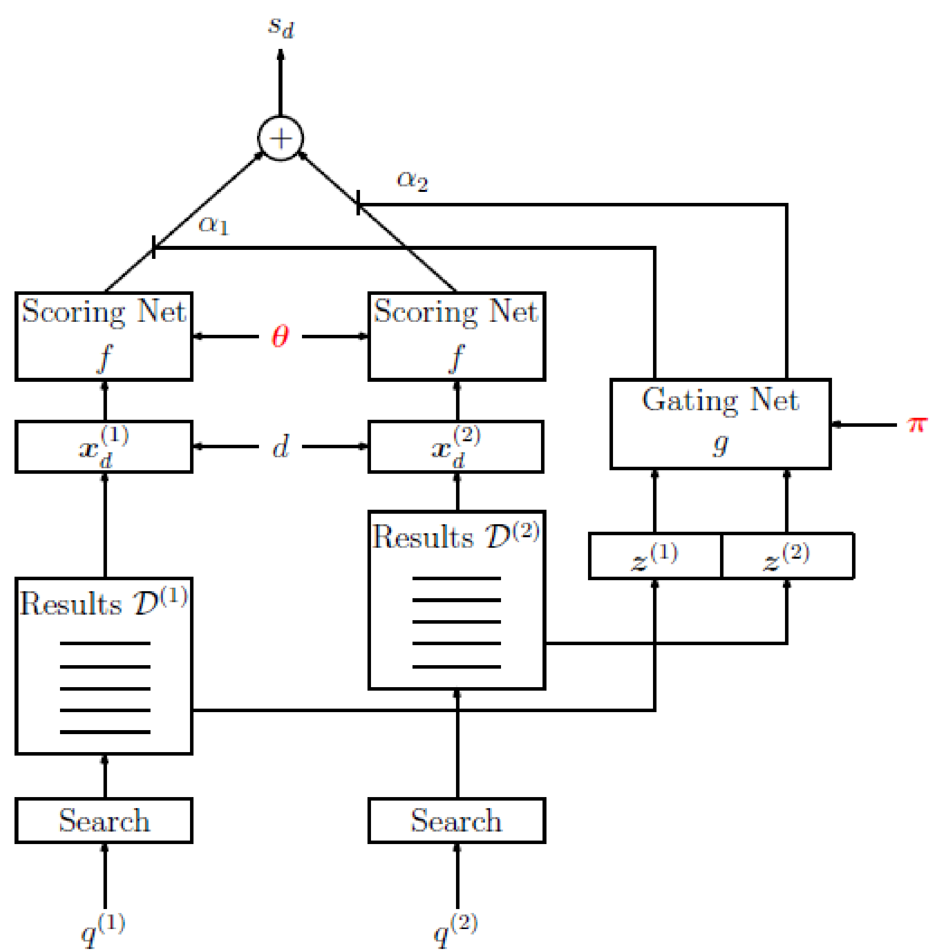
- Widely used in information retrieval

Learning to Rank (Sheldon et al., 2011)

- LambdaMerge: learning a single model for matching and ranking
- LambdaRank as ranker
- Directly optimizing NDCG
- Features
  - Matching scores
  - Quality of reformulation
  - Quality of search result

$$s_d = \sum_k \alpha_k \cdot f(x_d^{(k)}; \theta).$$

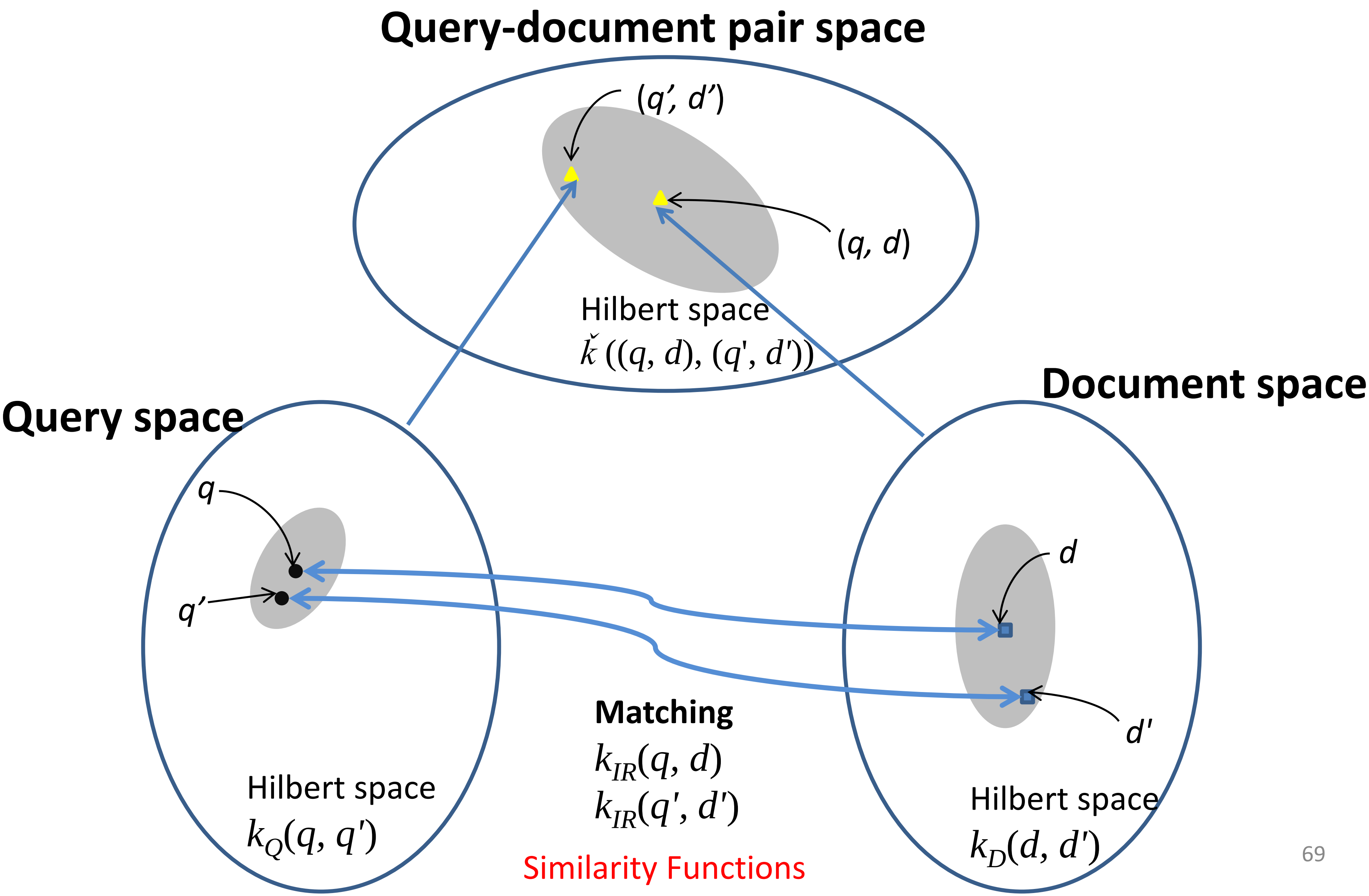
$$\alpha_k = \frac{\exp(\pi^T z^{(k)})}{\sum_p \exp(\pi^T z^{(p)})}$$



Kernel Method (Wu et al, 2011)

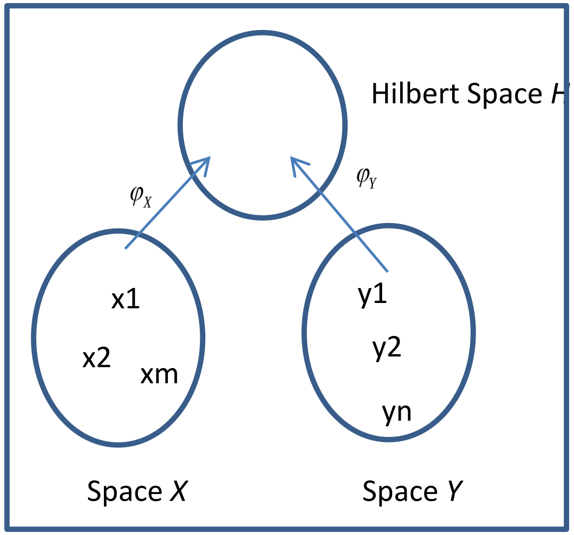
- Query similarity and document similarity are given
- ‘Smooth query document similarity’ by those of similar queries and documents
- Interpretation: nearest neighbor in space of query document pairs (double KNN)
- Automatically learning the weights of linear combination from click-data
- Theoretically sound approach

Kernel Method
(Wu et al, 2011)



Learning of Matching Model

- Matching Function : $k(x, y) = \langle \varphi_X(x), \varphi_Y(y) \rangle_{\mathcal{H}}$
- Input
  - Training data $S = \{(x_i, y_i), r_i\}_{1 \leq i \leq N}$
- Output
  - Matching Function
- Optimization



$$\min_{k \epsilon \mathcal{K}} \frac{1}{N} \sum_{i=1}^{N} l(k(x_i, y_i), r_i) + \Omega(k)$$

Learning of Matching Model Using Kernel Methods

- Assumption
  - Space of matching functions is RKHS generated by positive definite kernel $\bar{k}: (X \times Y) \times (X \times Y)$
- Optimization
  - $\min_{k \in K} \frac{1}{N} l(k(x_i, y_i), r_i) + \frac{\lambda}{2} \|k\|^2$
- Solution
  - $k^*(x, y) = \sum_{i=1}^{N} \alpha_i \bar{k}\,(x_i, y_i), (x, y))$

Learning Robust BM25

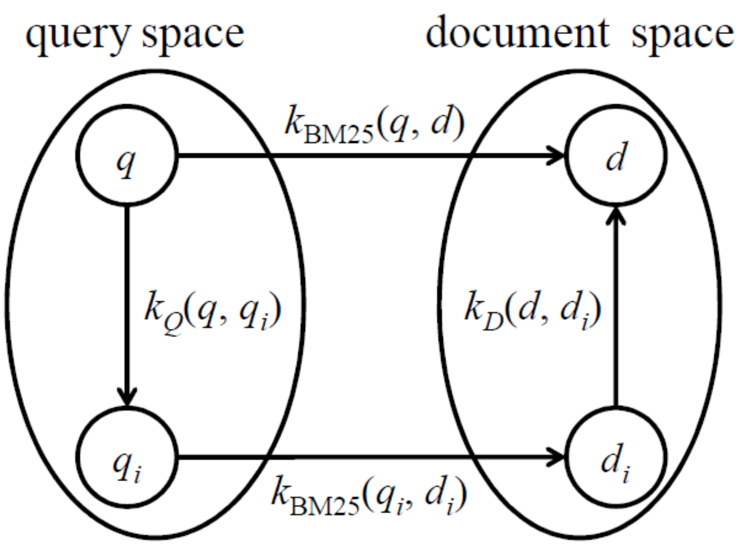
- BM25 :
- Kernel

$$\bar{k}\big((q,d),(q',d')\big) = k_{BM25}(q,d)k_Q(q,q')k_D(d,d')k_{BM25}(q',d')$$

- Solution (called Robust BM25)

$$k_{RBM25}(q,d) = k_{BM25}(q,d) \cdot \sum_{i=1}^{N} \alpha_i k_Q(q,q_i)k_D(d,d_i)k_{BM25}(q_i,d_i)$$

- Deal with term mismatch

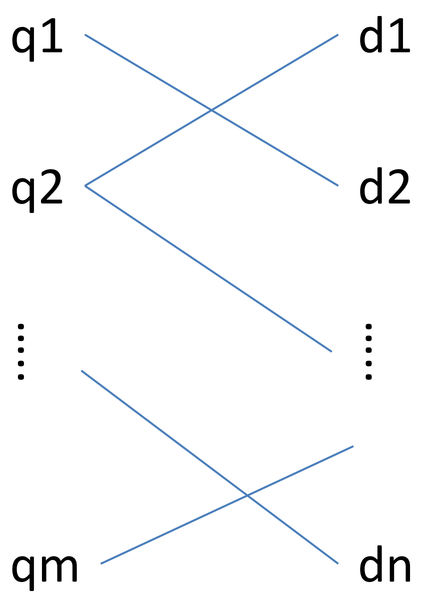


Similar Query Mining Problem

- Task
  - Given click-through data or search session data
  - Find similar queries or similar query patterns
  - E.g., ny → new york, distance between X and Y → how far is X from Y
- Challenge
  - Dealing with noises

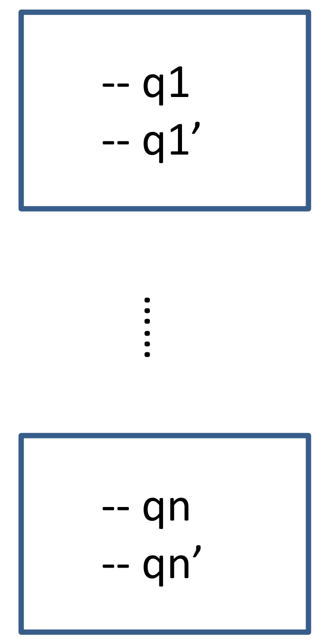
Mining of Similar Queries

Click-through data



Similar queries can be found by co-click

Search session data



Similar queries can be found from users' query reformulations

Methods of Similar Query Mining

- Using click-through data
  - Calculating Pearson correlation coefficient (Xu & Xu, '11)
  - Agglomerative clustering (Beeferman & Burger, '00), DBScan (Wen et al, '01), K-means (Baeza-Yates et al, '04), Query stream clustering (Cao et al, '08; Liao et al, '12)
  - Random walk (Craswell & Szummer, '07)
- Using search session data
  - Calculating Jacaard similarity (Huang et al, '03), mutual information (Jensen et al, '06), likelihood ratio (Jones et al, '06)
- Learning of query similarity
  - Query similarity learning as metric learning (Xu & Xu, '11)
- Learning of query reformulation patterns
  - Mining of natural language question patterns (Xue et al, '12)

Pearson Correlation Coefficient (Xu & Xu 2011)

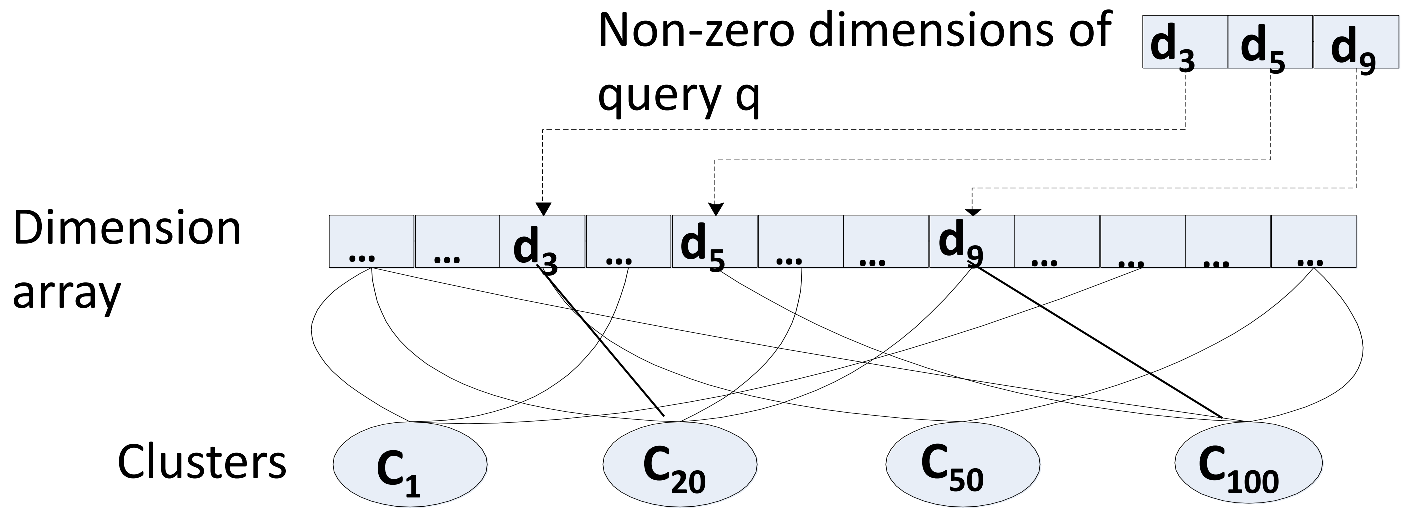
- Use click-through Bipartite Graph
- Assume that queries sharing many clicked URLs are similar
- One step random walk
- Measure

$$r = \frac{\sum_{i=1}^{n}(q_i - \overline{q})(p_i - \overline{p})}{\sqrt{\sum_{i=1}^{n}(q_i - \overline{q})^2}\sqrt{\sum_{i=1}^{n}(p_i - \overline{p})^2}}$$

- Selecting queries having large PCC values

Query Stream Clustering (Cao et al, 2011; Liao et al, 2012)

- Average time complexity: linear order
- Each query has only 3.1 clicked URLs, each URL has only 3.7
- Only non-zero elements matter when using cosine similarity
- Dimension array



Query Stream Clustering

- An element at dimension array links to clusters having vectors with non-zero values at this element
- Algorithm
  - Create cluster for first query
  - Repeat
    - If current query is close to one of the existing clusters, assign it to the cluster
    - Similarity calculation using dimension array (very efficient)
    - Otherwise, create new cluster for current query
  - Post processing to refine clusters

Random Walk (Craswell & Szummer, 2007)

- Transition probability

$$P_{t+1|t}(j \mid i) = \begin{cases} (1-s)\dfrac{C_{ij}}{\sum_k C_{ik}} & \text{when } i \neq j \\ s & \text{when } i = j \end{cases}$$

- Large self transition probability (query is similar to itself)
- Random Walk

$$P_{t|0}(j \mid i) = [\mathbf{A}^t]_{ij}$$

Likelihood Ratio Testing (Jones et al. 2006)

- Testing the hypothesis that seeing Q_b is independent of seeing Q_a
  - $H_1: P(Q_b \mid Q_a) = p = P(Q_b \mid \neg Q_a)$
  - $H_2: P(Q_b \mid Q_a) = p_1 \neq p_2 = P(Q_b \mid \neg Q_a)$
  - Log Likelihood Ratio: $LLR = -2 \log \frac{L(H_1)}{L(H_2)}$
- Suppose the data follows binomial distribution, then LLR follows χ^2 distribution
  - If LLR>3.84, then 95% confidence to reject the H_1 hypothesis

Query Similarity Learning as Metric Learning (Xu & Xu, '11)

- Similar query pairs and dissimilar query pairs are given
- Can we learn from head and propagate it to tail?
- From fact "hotmail sign up" are "hotmail sign on" similar to learn fact "X sign up" and "X sign on" are similar

Query Similarity Learning

- Objective function

$$\max_{M \succeq 0} \sum_{(q_i,q_j)\in S_+} \frac{\phi(q_i)^T M \phi(q_j)}{\sqrt{\phi(q_i)^T M \phi(q_i)}\sqrt{\phi(q_j)^T M \phi(q_j)}} - \sum_{(q_i,q_j)\in S_-} \frac{\phi(q_i)^T M \phi(q_j)}{\sqrt{\phi(q_i)^T M \phi(q_i)}\sqrt{\phi(q_j)^T M \phi(q_j)}} - \lambda \|M\|_1$$

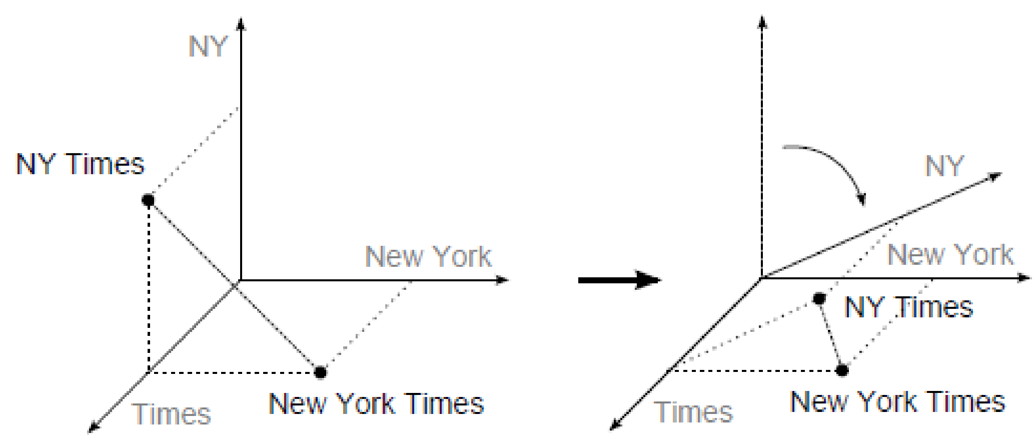
- Efficient optimization algorithm

Query Similarity Learning

- N-gram vector space
- Similar query pairs and dissimilar query pairs are given
- Dot product as similarity
- Learning linear transformation (weighted dot product)

$$sim(\phi(q_i), \phi(q_i)) = \frac{\phi(q_i)^T M \phi(q_j)}{\sqrt{\phi(q_i)^T M \phi(q_i)}\sqrt{\phi(q_j)^T M \phi(q_j)}},$$

- *M*: positive semi-definite



Mining of Natural Language Question Patterns (Xue et al. 2012)

- Steps
  - Collect query pairs from session data where first query is 5w1h question
  - Remove common words (except stopwords) in query pairs to create query reformulation patterns
  - Output high frequency patterns
- Example pattern

Table 2: Question reformulation patterns generated for the query pair ("how far is it from Boston to Seattle" ,"distance from Boston to Seattle").

S_1 = {Boston}:("how far is it from X_1 to Seattle" ,"distance from X_1 to Seattle")
S_2 = {Seattle}:("how far is it from Boston to X_1" ,"distance from Boston to X_1")
S_3 = {Boston, Seattle}:("how far is it from X_1 to X_2" ,"distance from X_1 to X_2")

QRU-1 Dataset

Joint Work with Michael Bendersky,
Gu Xu, Bruce Croft

Downloadable at MSR Web Site

bit.ly/qru1dataset

Motivation for QRU-1

- Benchmark dataset for research on query reformulation, etc
- Queries are as real as possible
- Queries are related to existing benchmark datasets (e.g., TREC query sets) for better connection with existing work

Content of Dataset

- Seed: 100 queries from TREC Web Track (2009 and 2010)
- Each query is assigned similar queries (on average 20 queries)
- Similar queries represent the same or similar search intents as original queries
- Similar queries may contain typos, stemming, synonyms
- In total, 2036 similar queries

Examples of Similar Queries

1:obama family tree

barack obama family
obama family
obama s family
barack obama family tree
the obama family
barack obama s family
obamas
obama genealogy
barack obama s family tree
barack obama ancestry
president obama s family
obamas family
obama family history
obama s family tree
barack obama genealogy
barack obama family history
barack obama geneology
president obama and family
obama s ancestry
barak obama family tree
barak obama family
obama family tre
obama and family tree

Examples of Similar Queries

95: earn money at home

earn money from home
earn money at home
how to earn money at home
earn money on the internet
ways to earn money at home
how to earn money from home
earn extra money at home
earning money from home
earn extra cash at home
earning money at home
earn at home
earn money working from home
earn money from home free
how to earn money on the internet
earn cash at home
earn currency at home
earn money at hom
earn money at hoem

Process of Data Creation

- Obtained 100 TREC queries
- Trained a query generation model using the method by ***(Wang et al. 2011)*** and search log data at Bing (2010/07-2010/12)
- Generated similar queries from TREC queries with the model
- Manually removed mistakenly generated queries (23% of generated queries were removed)
- Observed about 70% of the generated queries actually exist in real Bing log data
- Got approval for release from MS legal team

Guidelines for Manual Cleaning

- Keep generated queries, if
  - they represent the same intents as the original queries, and
  - they are likely to be input by users, including typos
- Otherwise discard the queries
  - E.g. “pictures of the obama family”
  - E.g. “obama family plant”
  - E.g. “michelle obama family tree”

One Possible Way of Using The Data

- Assuming similar queries are submitted by users
- Conducting retrieval and ranking on TREC Web Track documents with the similar queries
- The relevance performance can be worse or better than original queries
- Conducting query transformations on the similar queries to improve the relevance performance

Query Reformulation using QRU-1

SD	MAP	NDCG@20	ERR@20
Baseline metric	19.13	20.19	8.34
Best metric	25.00 *(+30.7%)*	32.88 *(+62.9%)*	15.09 *(+80.9%)*
% outperforming queries	12%	16%	18%
% topics improved	51%	63%	67%

Query Reformulation using QRU-1

SD	MAP	NDCG@20	ERR@20
Baseline metric	19.13	20.19	8.34
Best metric	25.00 *(+30.7%)*	32.88 *(+62.9%)*	15.09 *(+80.9%)*
% outperforming queries	12%	16%	18%
% topics improved	51%	63%	67%

Only small fraction of query reformulations improve performance

Query Reformulation using QRU-1

SD	MAP	NDCG@20	ERR@20
Baseline metric	19.13	20.19	8.34
Best metric	25.00 *(+30.7%)*	32.88 *(+62.9%)*	15.09 *(+80.9%)*
% outperforming queries	12%	16%	18%
% topics improved	51%	63%	67%

However, for a large number of topics there is at least one good reformulation

Term Substitution

Topic Title #1	*obama family tree*	**ERR@20**	*13.42*
Reformulations	barack obama ancestry		32.93
	obama s family		32.40
	barack obama s family		32.05

Query Expansion

Topic Title #5	*mitchell college*	**ERR@20**	*1.2*
Reformulations	mitchell college new london		19.6
	mitchell college new london ct		19.2
	www mitchell edu		5.7

Abbreviation induction

Topic Title #44	*map united states*	**ERR@20**	*3.42*
Reformulations	map usa states		13.92
	map usa		10.34
	united states america map		7.33

References

- F. Ahmad and G. Kondrak. Learning a spelling error model from search query logs. In Proc. of EMNLP, 2005.
- Baeza-Yates, R. et al. Query Clustering for Boosting Web Page Ranking. AWIC'04.
- N. Balasubramanian, G. Kumaran, and V.R. Carvalho. Exploring reductions for long web queries. In Proc. of SIGIR, 2010.
- Beeferman, D. and Berger, A.L. Agglomerative clustering of a search engine query log. KDD'00
- . Bergsma and Q. I. Wang. Learning noun phrase query segmentation. In Proc. of EMNLP-CoNLL, 2007.
- Eric Brill and Robert C. Moore. An improved error model for noisy channel spelling correction. In Proc. of ACL, 2000.
- Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, Hang Li: Context-aware query suggestion by mining click-through and session data. KDD 2008
- Huang, C.-K., et al. Relevant term suggestion in interactive web search based on contextual information in query session logs. Journal of the American Society for Information Science and Technology, 2003
- Q. Chen, M. Li, and M. Zhou. Improving query spelling correction using web search results. In Proc. of EMNLP-CoNLL, 2007.
- Silviu Cucerzan, Eric Brill: Spelling Correction as an Iterative Process that Exploits the Collective Knowledge of Web Users. In Proc. of EMNLP 2004.

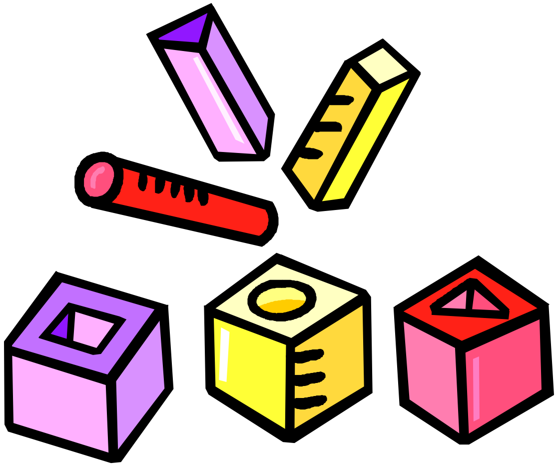
References

- F. Peng, N. Ahmed, X. Li, and Y. Lu. Context sensitive stemming for web search. In Proc. of SIGIR , 2007.
- K. M. Risvik, T. Mikolajewski, and P. Boros. Query segmentation for web search. In Proc. of WWW, 2003.
- W. Bruce Croft, Michael Bendersky, Hang Li, Gu Xu: Query representation and understanding workshop. SIGIR Forum 44(2), 2010.
- Duan, H. and Hsu, B.-J. P. Online spelling correction for query completion. In Proc. of WWW, 2011.
- Jiafeng Guo, Gu Xu, Hang Li, Xueqi Cheng. A Unified and Discriminative Model for Query Refinement. In Proc. of SIGIR, 2008.
- Jansen, B.J. and Spink, A. How are we searhing the world wide web? a comparison of nine search engine transaction logs. Information processing & management 2006
- R. Jones, B. Rey, O.Madani, and W. Greiner. Generating query substitutions, In Proc. of WWW, 2006.
- Hang Li, Gu Xu, W. Bruce Croft, Michael Bendersky, Ziqi Wang, Evelyne Viegas, QRU-1: A Public Dataset for Promoting Query Representation and Understanding Research, In Proceedings of the Workshop on Web Search Click Data, (WSCD'12), 2012.
- M. Li, M. Zhu, Y. Zhang, and M. Zhou. Exploring distributional similarity based models for query spelling correction. In Proc. of COLING-ACL, 2006.

References

- Zhen Liao, Daxin Jiang, Enhong Chen, Jian Pei, Huanhuan Cao, Hang Li: Mining Concept Sequences from Large-Scale Search Logs for Context-Aware Query Suggestion. ACM TIST 3(1): 17 (2011)
- Naoaki Okazaki, Yoshimasa Tsuruoka, Sophia Ananiadou, and Jun'ichi Tsujii. A discriminative candidate generator for string transformations. In Proc. of EMNLP, 2008.
- Daniel Sheldon, Milad Shokouhi, Martin Szummer, Nick Craswell: LambdaMerge: merging the results of query reformulations. In Proc. of WSDM, 2011.
- B. Tan and F. Peng. Unsupervised query segmentation using generative language models and Wikipedia. In Proc. of WWW, 2008.
- X. Wang and C. Zhai. Mining term association patterns from search logs for effective query reformulation. In Proc. of CIKM, 2008.
- Ziqi Wang, Gu Xu, Hang Li and Ming Zhang, A Fast and Accurate Method for Approximate String Search, In Proc. of ACL-HLT, 2011.
- Wen, J.-R., et al. Clustering user queries of a search engine. WWW' 01
- Wei Wu, Jun Xu, Hang Li, and Satoshi Oyama, Learning A Robust Relevance Model for Search Using Kernel Methods, JMLR, 2011.
- J. Xu & G. Xu, Learning Similarity Function for Rare Queries, Proc. Of WSDM 2011 X. Xue and W. B. Croft. Representing queries as distributions. In Proc. of SIGIR Workshop on Query Representation and Understanding, 2010.
- X. Xue, Y. Tao, D. Jiang, H. Li, Automatically Mining Question Reformulation Patterns from Search Log Data, Proc. of ACL 2012.

3. Matching with Dependency Model



Outline of Section 3

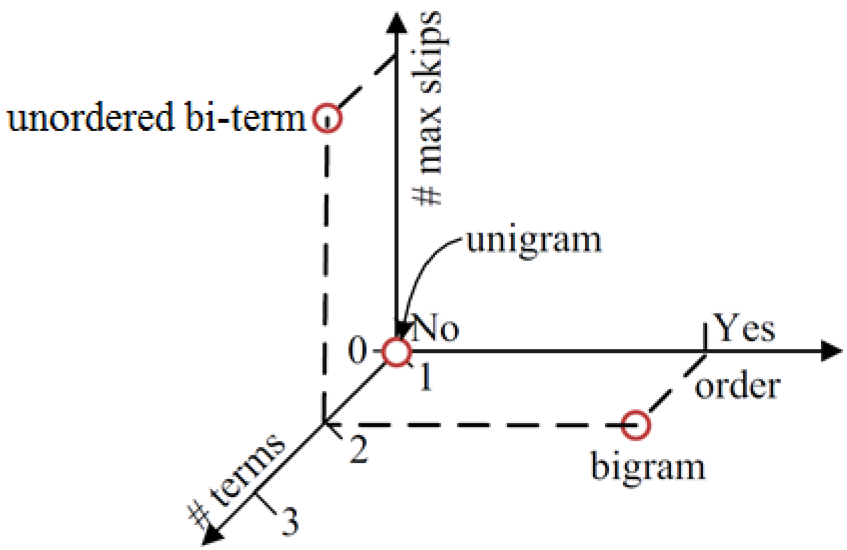
- Matching based on Term Dependency
- Term Dependency Models

Matching based on Term Dependency

- Matching of consecutive terms in query and document indicates higher relevance
  - “hot dog”
  - “hot dog” ≠ hot + dog
- Query: order is quite free, but not completely free
  - “hot dog recipe”, “recipe hot dog”
  - “hot recipe dog” ×
- Term dependency: a sequence of terms representing *soft* query segmentation

Factors of Term Dependency

- Number of terms
  - 1 term (unigram)
  - Multiple terms (bigram, bi-terms …)
- Order
  - N-gram
  - Unordered N-terms
- Number of max skips
  - No skip
  - S skips



Types of Term Dependency

- Term dependency in query
  - Noun phrases (Bendersky & Croft, ’08)
  - Phrases & proximities (Bendersky & Croft, ’10; Shi & Nie, ’10; Bendersky & Croft, ‘12)
- Latent term dependency
  - Pseudo relevance feedback (Cao et al., ’08; Metzler & Croft ’07; Lease ’08; Bendersky et al., ’11)
  - Query expansion (Metzler ’11)

Addressing Term Mismatch based on Term Dependency

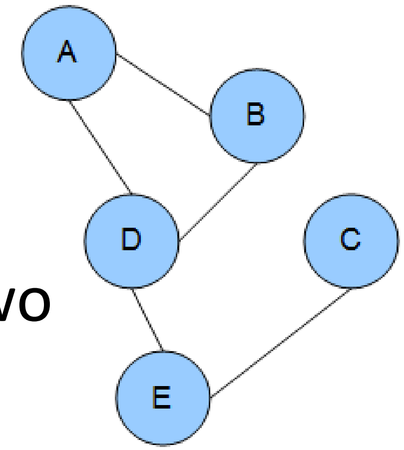
- Term dependency in query represents degree of matching between query and document
  - Document including “hot dog” has higher matching degree than document including “hot” and “dog”
- Latent term dependency uses relations with additional terms to help ‘infer’ degree of matching

Matching with Term Dependencies

- Term dependencies using Markov Random Fields (MRF)
  - Explicit term dependencies (Metzler & Croft, ’05)
  - Latent term dependencies (Metzler & Croft, 2008; Bendersky et al, ’11)
  - Weighted term dependencies (Bendersky et al., ’10)
- Higher-order term dependencies using query hypergraphics (Bendersky & Croft, ’12)
- Term dependencies using discriminative model (Shi & Nie, ’10)

Markov Random Fields

- Joint probability distribution represented by undirected graph
  - Nodes: random variables
  - Edges: dependencies between variables
  - Cliques: subset of nodes such that every two nodes are connected



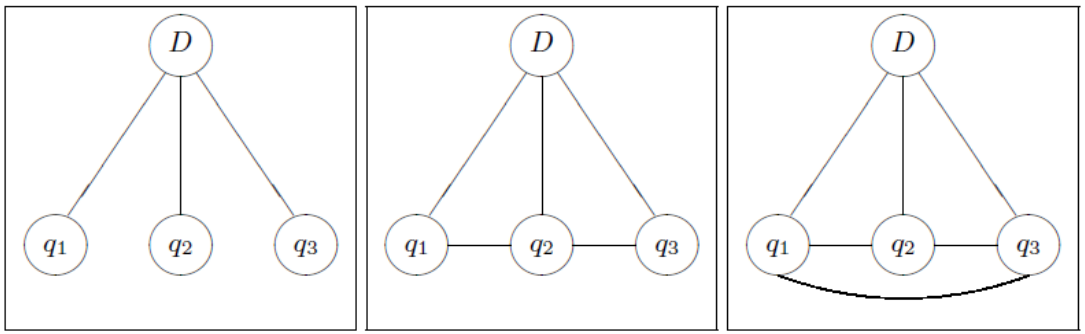
- Factorization of joint probability based on cliques

$$P(x_1 \cdots x_N) = \frac{1}{Z} \prod_{c \in clique(G)} \psi(c)$$

normalizing factor

potential function

Modeling Term Dependencies with MRF (Metzler & Croft, 2005)



independence sequential dependence full dependence

- Nodes
  - Document node
  - One node for each query term
- Edges
  - Each query node is linked with document node
  - Dependent terms are linked together

Modeling Term Dependencies with MRF

- Cliques
  - Representing how query terms are matched in document
  - Matching scores determined by potential function
- Joint probability

$$P_{\Lambda}(\mathbf{q}, \mathbf{d}) = \frac{1}{Z_{\Lambda}} \prod_{c \in clique(G)} \exp(\lambda_c f(c))$$

- Matching function

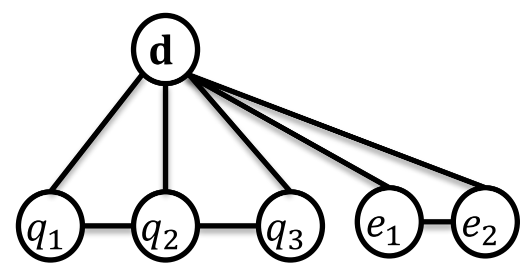
$$P(\mathbf{d}|\mathbf{q})$$

Modeling Term Dependencies with MRF

- Feature functions $f(c)$
  - Term:
    $$f_T(q_i, \mathbf{d}) = \log\left[(1-\alpha_{\mathbf{d}})\frac{tf_{q_i,\mathbf{d}}}{|\mathbf{d}|} + \alpha_{\mathbf{d}}\frac{cf_{q_i}}{|C|}\right]$$
  - Ordered phrase:
    $$f_O(q_i \cdots q_{i+k}, \mathbf{d}) =$$
    $$\log\left[(1-\alpha_{\mathbf{d}})\frac{tf_{\#1(q_i \cdots q_{i+k}),\mathbf{d}}}{|\mathbf{d}|} + \alpha_{\mathbf{d}}\frac{cf_{\#1(q_i \cdots q_{i+k})}}{|C|}\right]$$
  - Unordered phrase:
    $$f_U(q_i \cdots q_{i+k}, \mathbf{d}) =$$
    $$\log\left[(1-\alpha_{\mathbf{d}})\frac{tf_{\#\mathrm{uwN}(q_i \cdots q_{i+k}),\mathbf{d}}}{|\mathbf{d}|} + \alpha_{\mathbf{d}}\frac{cf_{\#\mathrm{uwN}(q_i \cdots q_{i+k})}}{|C|}\right]$$

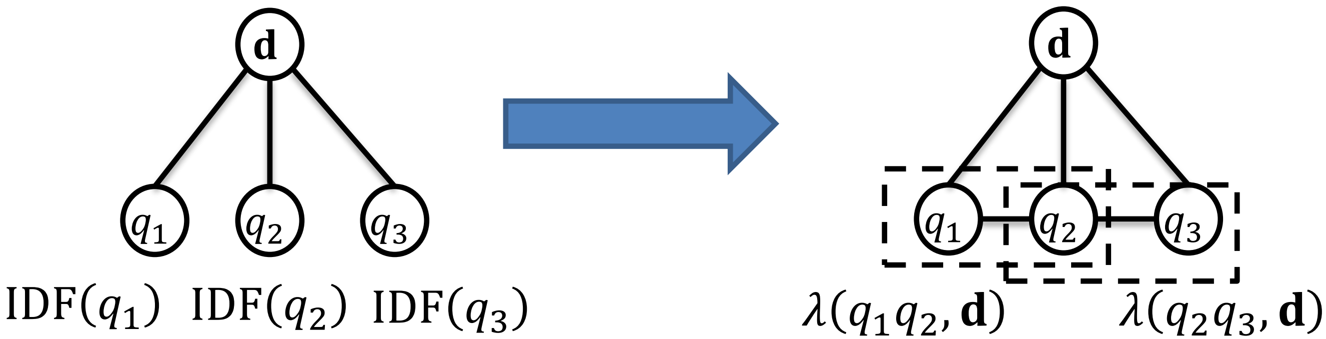
Latent Term Dependencies (Metzler & Croft, 2007)

- Assumption
  - Latent terms exist behind query
  - E.g., collecting terms by pseudo relevance feedback
- Modeling latent term dependencies
  - Constructing MRF on extended graph
  - Term dependencies between query $\mathbf{q}$ and document $\mathbf{d}$
  - Latent dependencies between $\boldsymbol{e} = e_1, \cdots, e_k$ and $\mathbf{d}$
  - Matching function $P(\mathbf{d}|\boldsymbol{q}, \boldsymbol{e})$



Utilizing and Learning Weights of Term Dependencies

- High weights for most discriminative term dependencies (like IDF for unigram)



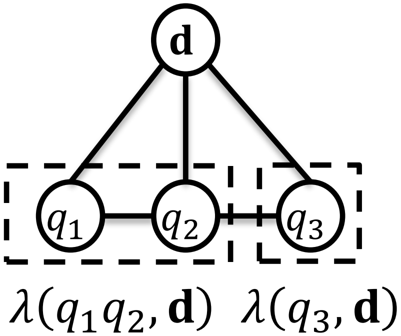
- Leveraging different data resources such as web N-gram, Wikipedia etc. for estimating weights

Weighted Term Dependencies (Bendersky et al., 2010)

- Represent $\lambda(c)$ with features

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

$$\lambda(q_i q_{i+1}, \mathbf{d}) = \sum_{j=1}^{k_{bi}} w_j^{bi} g_j^{bi}(q_i q_{i+1})$$



- Matching function

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

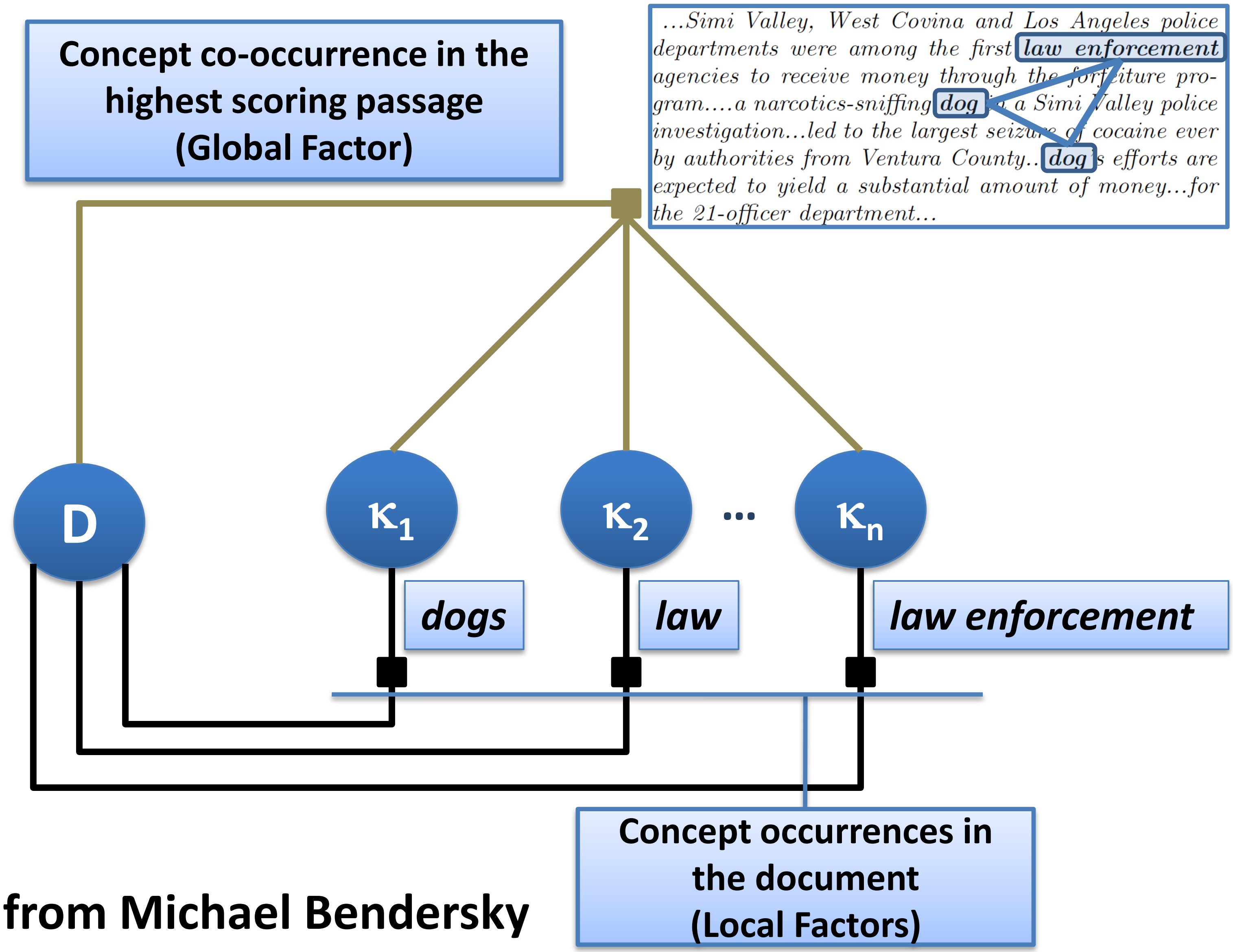
Features for Representing Weights

- Features from different data resources (e.g., web N-gram, query log, Wikipedia ...)

data source	feature	description
collection	cf_e df_e	collection frequency for *e* document frequency for *e*
N-Grams Query Log	$gf(e)$ $qe\_cnt(e)$ $qp\_cnt(e)$	n-gram count of *e* count of exact match of *e* and a query in the log count of times *e* occurs within a query in the log
Wikipedia titles	$we\_cnt(e)$ $wp\_cnt(e)$	Does *e* appears as a Wikipedia title? Count of times *e* occurs within a Wikipedia title

e can be either a query term q_i or a sequential query term pair $q_i q_{i+1}$

Query Hypergraphics for Dependencies (Bendersky & Croft, 2012)



Courtesy from Michael Bendersky

Discriminative Model for Dependency (Shi & Nie, 2010)

- Discriminative model

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

- Features are flexible

$$\begin{aligned} SC(D,Q) &= \sum_{q_i \in Q}\lambda_U(q_i|Q)f_U(q_i,D) \\ &+ \sum_{q_i q_{i+1}\in Q}\lambda_B(q_i q_{i+1}|Q)f_B(q_i,D) \\ &+ \sum_{w\in W}\sum_{q_i,q_j\in Q;i\neq j}\lambda_{C_w}\left(q_i,q_j|Q\right)f_{c_w}\left(q_i,q_j,D\right) \end{aligned}$$

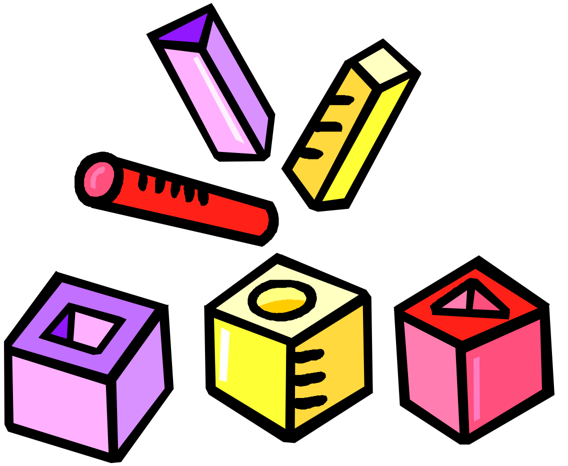
References

- Michael Bendersky and W. Bruce Croft. Discovering Key Concepts in Verbose Queries. In Proc. of SIGIR 2008.
- Michael Bendersky, Donald Metzler, and W. Bruce Croft. Learning Concept Importance using a Weighted Dependence Model. In Proc. of WSDM 2010.
- Michael Bendersky, Donald Metzler, and W. Bruce Croft. Parameterized Concept Weighting in Verbose Queries. In Proc. of SIGIR 2011.
- Michael Bendersky and W. Bruce Croft. Modeling higher-order Term Dependencies in Information Retrieval using Query Hypergrahics. In Proc. of SIGIR 2012.
- Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson: Selecting Good Expansion Terms for Pseudo-Relevance Feedback. In Proc. of SIGIR 2008.
- Van Dang, Michael Bendersky, and W. Bruce Croft. Learning to Rank Query Reformulations. In Proc. of SIGIR 2010.
- Hao Lang, Donald Metzler, Bin Wang, and Jin-Tao Li. Improved Latent Concept Expansion using Hierarchical Markov Random Fields. In Proc. of CIKM 2010.
- Matthew Lease, James Allan, and Bruce Croft. Regression Rank: Learning to Meet the Opportunity of Descriptive Queries. In Proc. of ECIR 2009.
- Matthew Lease. Incorporating Relevance and Pseudo-relevance Feedback in the Markov Random Field Model. In Proc. of TREC 2008.

References

- Matthew Lease. An Improved Markov Random Field Model for Supporting Verbose Queries. In Proc. of SIGIR 2009.
- Donald Metzler and W. Bruce Croft. A Markov Random Field Model for Term Dependencies. In Proc. of SIGIR 2005.
- Donald Metzler and W. Bruce Croft. Linear Feature-based Models for Information Retrieval. Information Retrieval, 2006.
- Donald Metzler and W. Bruce Croft. Latent Concept Expansion using Markov Random Fields. In Proc. of SIGIR 2008.
- Donald Metzler. Feature-based Query Expansion. In Proc. of SIGIR 2011.
- Lixin Shi and Jian-Yun Nie. Using Various Term dependencies according to Their Utilities. In Proc. of CIKM 2010.
- Krysta M. Svore, Pallika H. Kanani, and Nazan Khan. How Good is a Span of Terms? Exploiting Proximity to Improve Web Retrieval. In Proc. of SIGIR 2010.
- Lidan Wang, Donald Metzler, and Jimmy Lin. Ranking under Temporal Constraints. In Proc. of CIKM 2010.
- Jun Xu, Hang Li, and Chaoliang Zhong. Relevance Ranking using Kernels. In Proc. of AIRS 2010.
- Le Zhao and Jamie Callan. Term Necessity Prediction. In Proc. of CIKM 2010.

4. Matching with Statistical Machine Translation



Outline of Section 4

- Statistical Machine Translation
- Matching with Translation Model
- Issues in Matching with Translation Model
- Methods for Matching with Translation Models

Statistical Machine Translation (SMT)

- Given sentence C in source language, translates it into sentence E in target language

$$E^* = \text{argmax}_E P(E|C)$$

- Linear combination of features

$$P(E|C) = \frac{1}{Z(C,E)} \exp \sum_i \lambda_i h_i(C,E)$$

$$E^* = \text{argmax}_E \sum_i \lambda_i h_i(C,E)$$

Typical Translation Models

- Word-based
  - Translating word to word
- Phrase-based
  - Translating based on phrase
- Syntax-based
  - Translating based on syntactic structure

Word-based Model: IBM Model One (Brown et al., 1993)

C: <NULL> 这 房子 很 小

$P(the|这)$

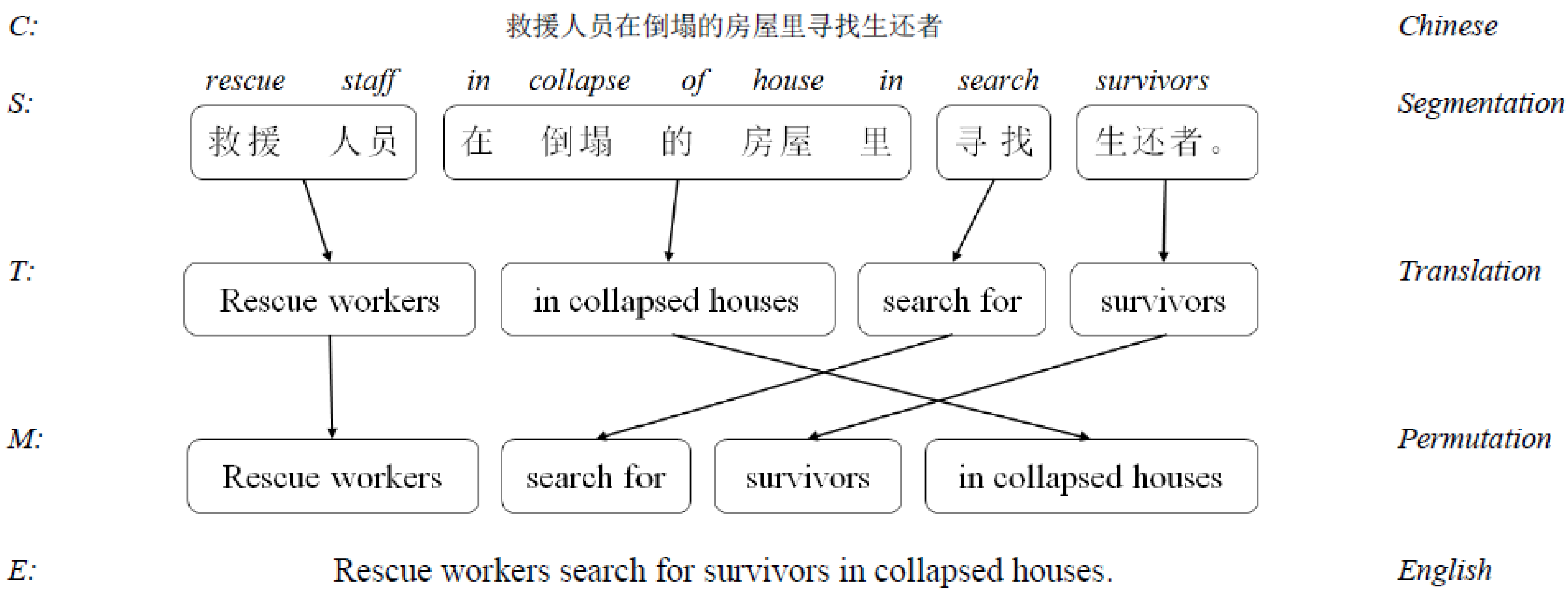
E: the house is very small

1 2 3 4 5

- Generating target sentence
  - Length M of target sentence is generated
  - For each target sentence position, $i = 1\text{: M}$
    - Word c_j in source sentence C is selected
    - e_i at position i is generated depend on c_j

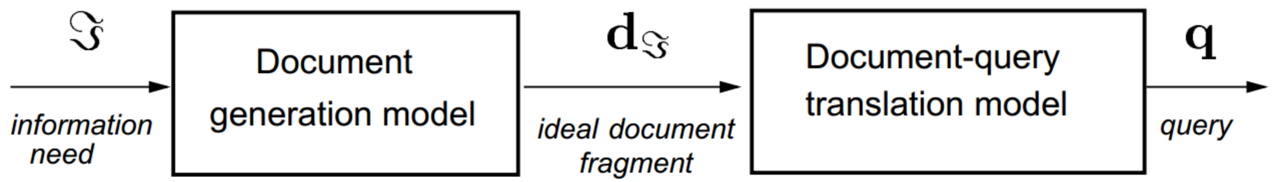
$$P(E|C) = \frac{\epsilon}{(L+1)^M} \prod_{i=1}^{M} \sum_{j=1}^{N} P(e_i|c_j)$$

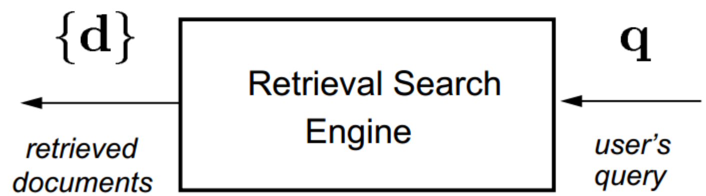
Phrase-Based Models



Courtesy from Jianfeng Gao

Model of Query Generation and Retrieval





- Task of retrieval: find the a posteriori most likely documents given query

$$P(\mathbf{d}|\mathbf{q},\mathcal{U}) = \frac{P(\mathbf{q}|\mathbf{d},\mathcal{U}) \cdot P(\mathbf{d}|\mathcal{U})}{P(\mathbf{q}|\mathcal{U})}$$

query dependent

query independent

Matching with Translation Model

- Translating document **d** to query **q** (or translation document language model to query language model)
- Given query **q** and document **d**, translation probability is viewed as matching score between **q** and **d**
- Difference from conventional translation model
  - Translation in same language
  - Self translation plays important role

Addressing Term Mismatch with Translation Model

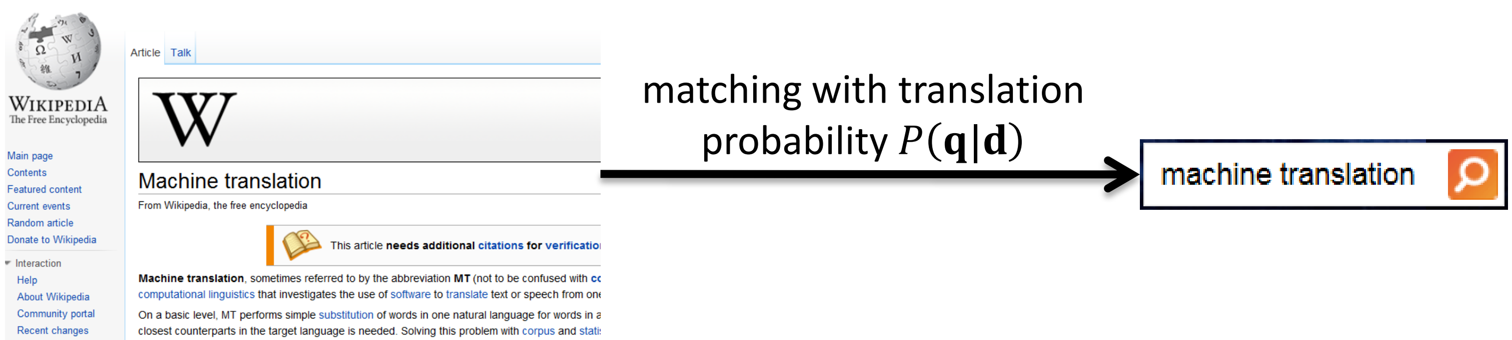
- Translation probability $P(q|w)$ represents matching degree between words in query and document

q	$P(q\|w)$	q	$P(q\|w)$
titanic	0.56218	Vista	0.80575
ship	0.01383	Windows	0.05344
movie	0.01222	Download	0.00728
pictures	0.01211	ultimate	0.00571
sink	0.00697	xp	0.00355
facts	0.00689	microsoft	0.00342
photos	0.00533	bit	0.00286
rose	0.00447	compatible	0.00270
people	0.00441	premium	0.00244
survivors	0.00369	free	0.00211

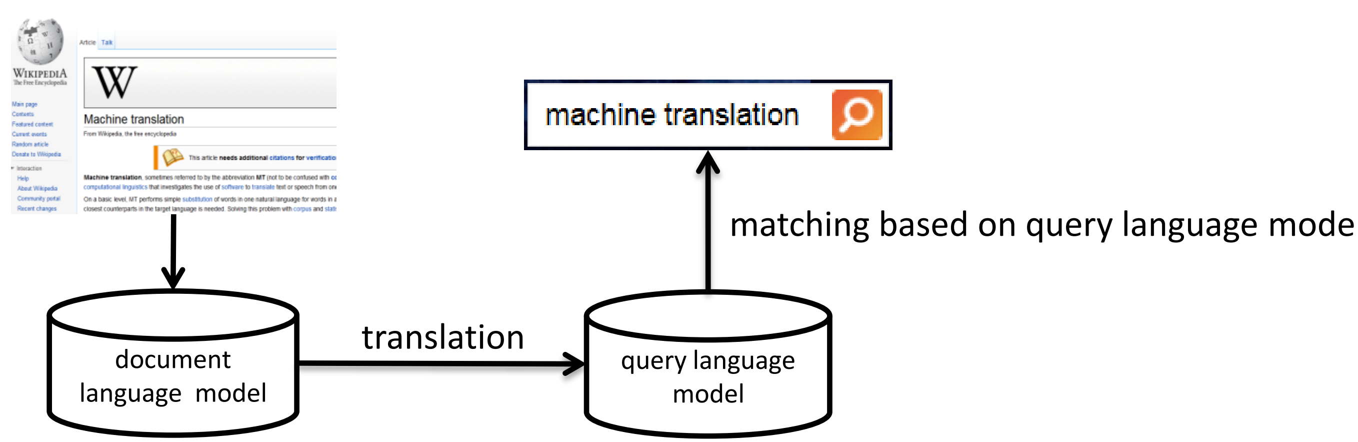
w = titanic　　　　w = vista

Approaches to Matching with Translation Model

- Translating document to query



- Translating document model to query model



Issues in Matching with Translation Models

- Types of Training Data
- Types of Document Fields
- Types of Translation Models

Types of Training Data for Learning Translation Probabilities

- Synthetic data (Berger & Lafferty, '99)
- Document collection (Karimzadehgan & Zhai, '10)
- Title-body pairs of documents (Jin et al., '02)
- Query-title pairs in click-through data (Gao et al., '10)

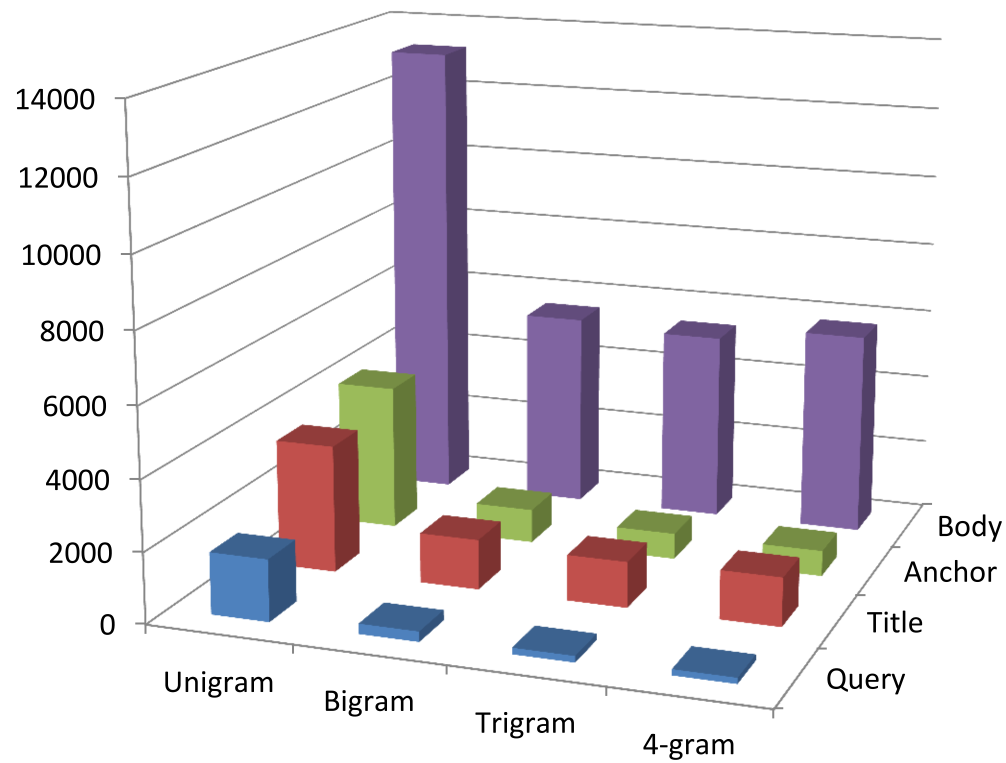
http://webmessenger.msn.com
title: "msn web messenger"

clicked queries	score
msn web	0.6675
webmensseger	0.6621
msn online	0.6403
Windows web messanger	0.6321
talking to friends on msn	0.6130
… …	

Types of Document Fields

- Use of title is better than body (Huang et al., '10)
- Titles and queries have similar languages
- Bodies and queries have very different languages

$$Perplexity(\tilde{P}, Q) = 2^{H(\tilde{P},Q)}$$
$$= 2^{-\sum_s \tilde{p}_s \log q_s}$$



Methods for Matching with Translation Models

- Translating document to query
  - Word-based model (Berger & Lafferty, ’99)
  - Phrase-based model (Gao et al., ’10)
  - Topic-based model (Gao et al., ’11)
  - Learning translation probabilities from documents (Karimzadehgan & Zhai, ’10)
- Translating document model to query model
  - Translated query language model (Jin et al., ’02)

Matching with Word-based Translation Model

- Basic model

$$P(\mathbf{q}|\mathbf{d}) = \prod_{q \in \mathbf{q}} P(q|\mathbf{d}) = \prod_{q \in \mathbf{q}} \sum_{w \in \mathbf{d}} \underbrace{P(q|w)}_{\text{translation probability}} \underbrace{P(w|\mathbf{d})}_{\text{document language model}}$$

- Smoothing to avoid zero translation probability (Berger & Lafferty, ’99)

$$P(\mathbf{q}|\mathbf{d}) = \prod_{q \in \mathbf{q}} \left(\alpha \underbrace{P(q|coll)}_{\text{background unigram model}} + (1-\alpha) \sum_{w \in \mathbf{d}} P(q|w) P(w|\mathbf{d}) \right)$$

- Adding self-translation (Gao et al., ’10)

$$P(\mathbf{q}|\mathbf{d}) = \prod_{q \in \mathbf{q}} \left(\alpha P(q|coll) + (1-\alpha) \left(\beta \underbrace{P(q|\mathbf{d})}_{\text{unsmoothed document model}} + (1-\beta) \sum_{w \in \mathbf{d}} P(q|w) P(w|\mathbf{d}) \right) \right)$$

Examples of Translation Probabilities

q	$t(q \mid w)$
solzhenitsyn	0.319
citizenship	0.049
exile	0.044
archipelago	0.030
alexander	0.025
soviet	0.023
union	0.018
komsomolskaya	0.017
treason	0.015
vishnevskaya	0.015

$w =$ solzhenitsyn

q	$t(q \mid w)$
carcinogen	0.667
cancer	0.032
scientific	0.024
science	0.014
environment	0.013
chemical	0.012
exposure	0.012
pesticide	0.010
agent	0.009
protect	0.008

$w =$ carcinogen

q	$t(q \mid w)$
zubin_mehta	0.248
zubin	0.139
mehta	0.134
philharmonic	0.103
orchestra	0.046
music	0.036
bernstein	0.029
york	0.026
end	0.018
sir	0.016

$w =$ zubin

q	$t(q \mid w)$
pontiff	0.502
pope	0.169
paul	0.065
john	0.035
vatican	0.033
ii	0.028
visit	0.017
papal	0.010
church	0.005
flight	0.004

$w =$ pontiff

q	$t(q \mid w)$
everest	0.439
climb	0.057
climber	0.045
whittaker	0.039
expedition	0.036
float	0.024
mountain	0.024
summit	0.021
highest	0.018
reach	0.015

$w =$ everest

q	$t(q \mid w)$
wildlife	0.705
fish	0.038
acre	0.012
species	0.010
forest	0.010
environment	0.009
habitat	0.008
endangered	0.007
protected	0.007
bird	0.007

$w =$ wildlife

Matching with Phrase-based Translation Models (Gao et al., '10)

- Phrase-based translation model

d:	… cold home remedies …	*title*
S:	["cold", "home remedies"]	*segmentation*
T:	["stuffy nose", "home remedy"]	*translation*
M:	(1 → 2, 2→ 1)	*permutation*
q:	"home remedy stuffy nose"	*query*

  - Maximum approximation

$$P(\mathbf{q}|\mathbf{d}) \approx \max_{(S,T,M)\in \boldsymbol{B}(\mathbf{q},\mathbf{d})} P(T|\mathbf{d},S)P(M|\mathbf{d},S,T)$$

  - Max probability assignment via dynamic programming

$$P(\mathbf{q}|\mathbf{d}) \approx \max_{(S,T,M)\in B(\mathbf{d},\mathbf{q},A^*)} P(T|\mathbf{d},S) = \max_{(S,T,M)\in B(\mathbf{d},\mathbf{q},A^*)} \prod_{k=1\ldots K} P(\mathbf{q}_k|\mathbf{w}_k)$$

Example of Translation Probabilities

q	$P(\mathbf{q}\|\mathbf{w})$	**q**	$P(\mathbf{q}\|\mathbf{w})$
titanic	0.43195	sierra vista	0.61717
rms titanic	0.03793	sv	0.02260
titanic sank	0.02114	vista	0.01678
titanic sinking	0.01695	sierra	0.01581
titanic survivors	0.01537	az	0.00417
titanic ship	0.01112	bella vista	0.00320
titanic sunk	0.00960	arizona	0.00223
titanic pictures	0.00593	dominoes sierra vista	0.00221
titanic exhibit	0.00540	dominos sierra vista	0.00221
ship titanic	0.00383	meadows	0.00029

w = rms titanic　　　**w** = sierra vista

Figure 6: Sample phrase translation probabilities learned from the word-aligned query-title pairs.

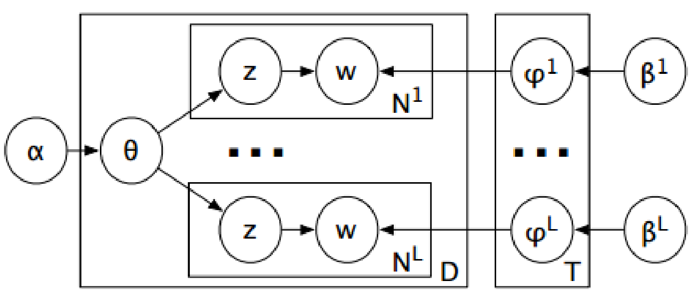
Improving Relevance

#	Models	NDCG@1	NDCG@3	NDCG@10
1	**BM25**	0.3181	0.3413	0.4045
2	**WTM_M1**	0.3310	0.3566	0.4232
3	**PTM (*l*=5)**	0.3355	0.3605	0.4254
4	**PTM (*l*=3)**	0.3349	0.3602	0.4253
5	**PTM (*l*=2)**	0.3347	0.3603	0.4252

Table 5: Ranking results on the evaluation data set, where only the title field of each document is used. **PTM** is the linear ranking model of Equation (22), where all the features, including the two phrase translation model features f_{PT} and f_{LW} (with different maximum phrase length, specified by l), are incorporated.

Polylingual Topic Model (Mimno et al., 2009)

- An extension of LDA
  - Modeling polylingual document tuples
  - Document tuple: documents that are loosely equivalent but written in different languages
  - E.g., Wikipedia articles in French, English and German.

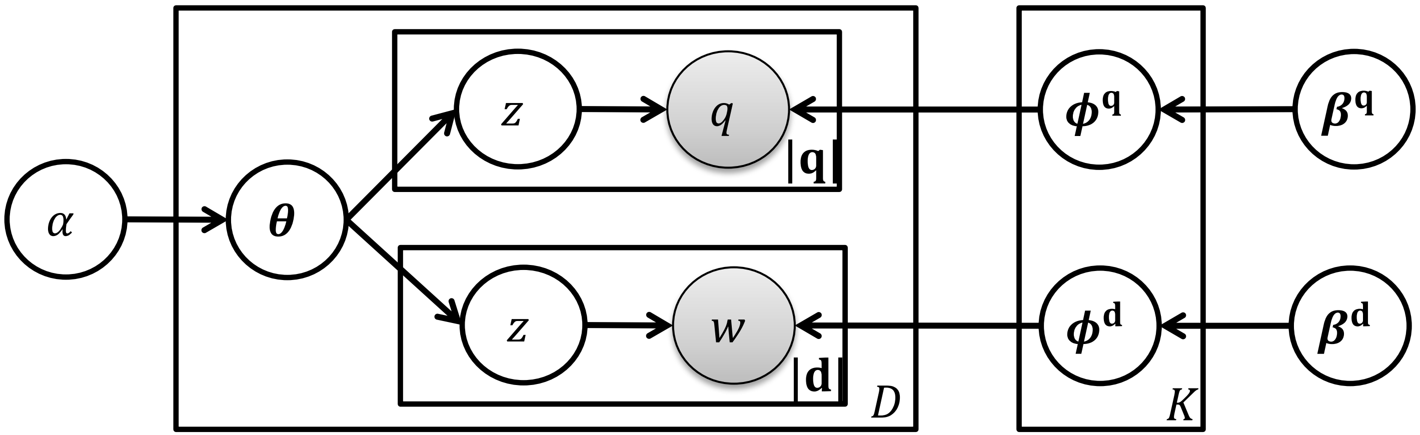


(a) Graphical model for PLTM.

Lang	%	Most probable words
Welsh	.20	ei greu swltan ottoman strwythr bymtheg
German	.18	osmanischen osmanische osmanen sultan konstantinopel truppen
Greek	.29	οθωμανική πόλης αυτοκρατορία μωάμεθ κωνσταντινούπολη
English	.15	ottoman the empire turkish ottomans constantinople
Finnish	.06	balkanin turkin muureja kaupungin toukokuuta tuottanut
French	.10	lempire ottoman sultan turcs ottomans éd
Italian	.07	turchi ottomano ottomani limpero sultano veneziano
Polish	.31	turcy turków murów rogu sułtan mury
Portug.	.18	turcos sultão constantinopla ataque otomano muralhas
Russian	.57	османской турки империи турок султан султана

(b) Top words for a single topic in ten languages, along with the percentage of each corpus assigned to this topic.

Topic-based Translation Model (Gao et al., 2011)



- Query and document use different vocabularies to express the same distribution of topics

$$P(\mathbf{q}|\mathbf{d}) = \prod_{q \in \mathbf{q}} P_{bltm}(q|\mathbf{d}) = \prod_{q \in \mathbf{q}} \sum_{z} P\left(q|\phi_z^{\mathbf{q}}\right) P\left(z|\theta^{\mathbf{d}}\right)$$

- Smoothing and addressing self translation

$$P_s(\mathbf{q}|\mathbf{d}) = \prod_{q \in \mathbf{q}} (\lambda_1 P(q|C) + (1-\lambda_1)(\lambda_2 P(q|\mathbf{d}) + (1-\lambda_2) P_{bltm}(q|\mathbf{d})))$$

unsmoothed background model

unsmoothed document model

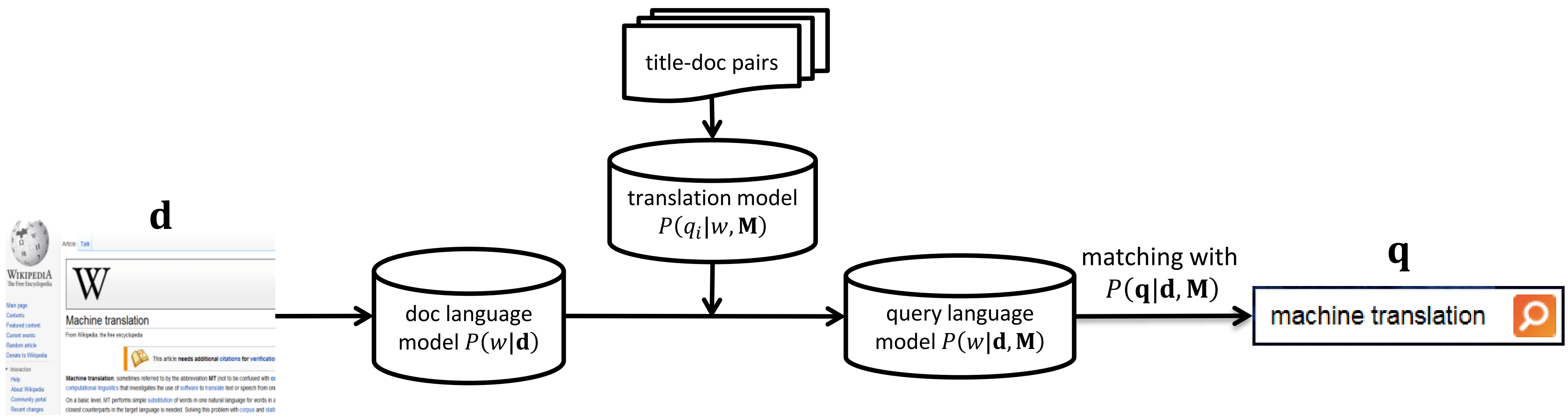
Improving Relevance

$\lambda_2 = 0$: no self-translation

#	Models	NDCG@1	NDCG@3	NDCG@10
1	UM	0.308	0.373	0.454
2	PLSA ($\lambda_2 = 0$)	0.295	0.371	0.456
3	PLSA	0.325	0.391	0.470
4	BLTM ($\lambda_2 = 0$)	0.330	0.399	0.476
5	BLTM	0.338	0.404	0.479
6	BLTM-PR ($\lambda_2 = 0$)	0.334	0.403	0.479
7	BLTM-PR	0.342	0.406	0.482
8	BLTM-PR-1V ($\lambda_2 = 0$)	0.337	0.403	0.480
9	BLTM-PR-1V	0.344	0.407	0.483
10	WTM_M1 ($\lambda_2 = 0$)	0.332	0.400	0.478
11	WTM_M1	0.338	0.404	0.480

Table 1: Web document ranking results using different topic models, tested on the evaluation data set, where only the title field of each document is used.

Matching with Translated Query Language Model (Jin et al., ’02)



$$P(\mathbf{q}|\mathbf{d},\mathbf{M}) = \epsilon \prod_{q_i \in \mathbf{q}} \lambda \left(\frac{P(q_i|\phi,\mathbf{M})}{|\mathbf{d}|+1} + \sum_{w \in \mathbf{d}} P(q_i|w,\mathbf{M}) P(w|\mathbf{d}) \right) + (1-\lambda) P(q_i|GE)$$

translate doc word to query word

document language model

background language model

Learning Translation Probabilities from Documents (Karimzadehgan & Zhai, '10)

- Mutual information of words (w, u)

$$I(w;u) = \sum_{X_w=0,1} \sum_{X_u=0,1} p(X_w, X_u) \log \frac{p(X_w, X_u)}{p(X_w)p(X_u)}$$

	$X_w = 0$	$X_w = 1$
$X_u = 0$		
$X_u = 1$		

- Translation probability

$$P_t(w|u) = \begin{cases} (1-\alpha)\dfrac{I(w;u)}{\sum_{w'} I(w';u)} & w \neq u \\ \alpha + (1-\alpha)\dfrac{I(u;u)}{\sum_{w'} I(w';u)} & w = u \end{cases}$$

Axiomatic Analysis of Translation Probabilities (Karimzadehgan & Zhai, '12)

- General constraints
  - Constraint 1: $\forall v, w, P(w|w) = P(v|v)$
  - Constraint 2: $\forall v, w, if\ w \neq v, then\ P(w|w) \geq P(w|v)$
  - Constraint 3: $\forall v, w, if\ w \neq v, then\ P(w|w) \geq P(v|w)$
- Additional constraints
  - Constraint 4: $if\ c(w,u) > c(w,v)\ and\ \sum_{w'} c(w',u) = \sum_{w'} c(w',v),$ $then\ P(w|u) > P(w|v)$
  - Constraint 5: $if\ c(w,u) = c(w,v)\ and\ \sum_{w'} c(w',u) < \sum_{w'} c(w',v),$ $then\ P(w|u) > P(w|v)$

  *$c(w,u)$: the number of co-occurrences of words $w\ and\ u$ in context

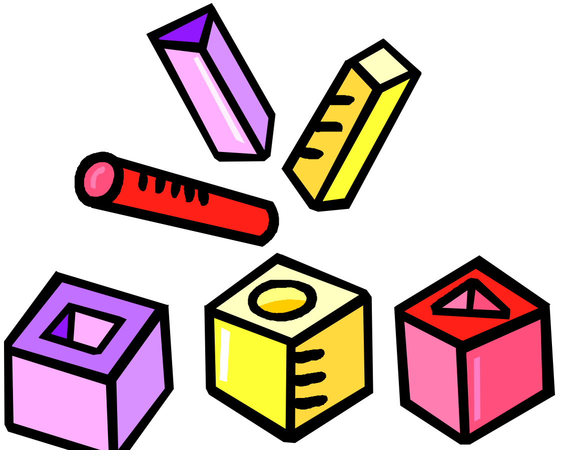
References

- Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. Bridging the Lexical Chasm: Statistical Approaches to Answer-Finding. In SIGIR 2000.
- Adam Berger and John Lafferty. Information Retrieval as Statistical Translation. In SIGIR 1999.
- Jianfeng Gao, Xiaodong He, and JianYun Nie. Click-through-based Translation Models for Web Search: from Word Models to Phrase Models. In CIKM 2010.
- Jianfeng Gao, Kristina Toutanova, and Wen-tau Yih. Clickthrough-based latent semantic models for web search. In SIGIR 2011.
- Jianfeng Gao : Statistical Translation and Web Search Ranking. http://research.microsoft.com/en-us/um/people/jfgao/paper/SMT4IR.res.pptx
- Dustin Hillard, Stefan Schroedl, and Eren Manavoglu, Hema Raghavan, and Chris Leggetter. Imrpoved Ad Relevance in Sponsored Search. In WSDM 2010.
- Jian Huang, Jianfeng Gao, Jiangbo Miao, Xiaolong Li, Kuansan Wang, Fritz Behr, and C. Lee Giles. Exploring web scale language models for search query processing. In WWW 2010.
- Ea-Ee Jan, Shih-Hsiang Lin, and Berlin Chen. Translation Retrieval Model for Cross Lingual Information Retrieval. In AIRS 2010.
- Rong Jin, Alex G. Hauptmann, and Chengxiang Zhai. Title Language Model for Information Retrieval. In SIGIR 2002.

References

- Maryan Karimzadehgan and Chengxiang Zhai. Estimation of Statistical Translation Models based on Mutual Information for Ad Hoc Information Retrieval. In SIGIR 2010.
- David Mimno , Hanna M. Wallach , Jason Naradowsky , David A. Smith, Andrew McCallum. Polylingual topic models. In EMNLP 2009.
- Seung-Hoon Na and Hwee Tou Ng. Enriching Document Representation via Translation for Improved Monolingual Information Retrieval. In SIGIR 2011.
- Jae-Hyun Park, W. Bruce Croft, and David A. Smith. Qusi-Synchronous Dependence Model for Information Retrieval. In CIKM 2011.
- Stefan Riezler and Yi Liu. Query Rewritting Using Monolingual Statistical Machine Translation. In ACL 2010.
- Dolf Trieschnigg, Djoerd Hiemstra, Franciska de Jong, and Wessel Kraaij. A cross-lingual Framework for Monolingual Biomedical Information Retrieval. In CIKM 2010.
- Elisabeth Wolf, Delphine Bernhard, and Iryan Gurevych. Combining Probabilistic and Translation-based Models for Information Retrieval based on Word Sense Annotations. In CLEF Workshop 2009.

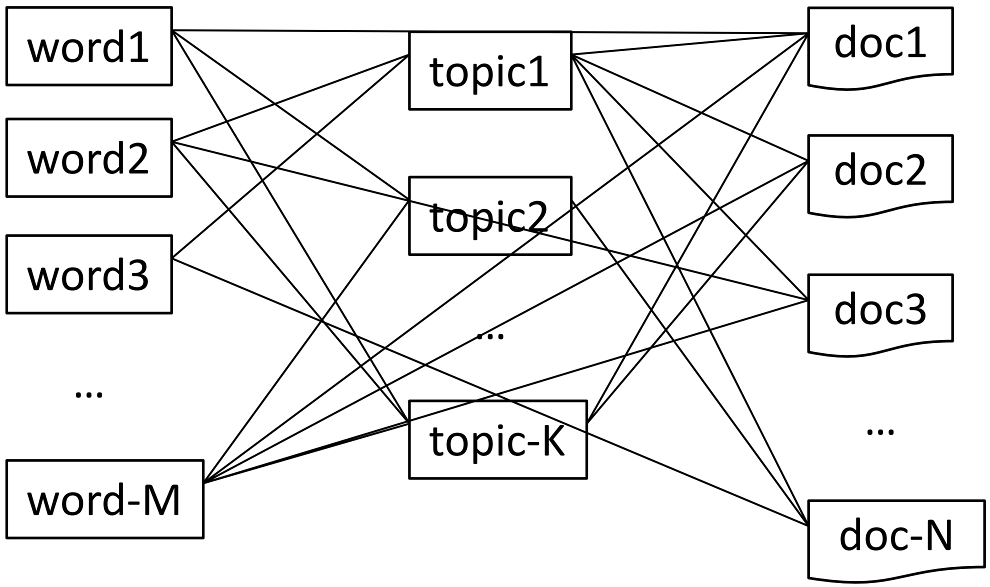
5. Matching with Topic Model



Outline of Section 5

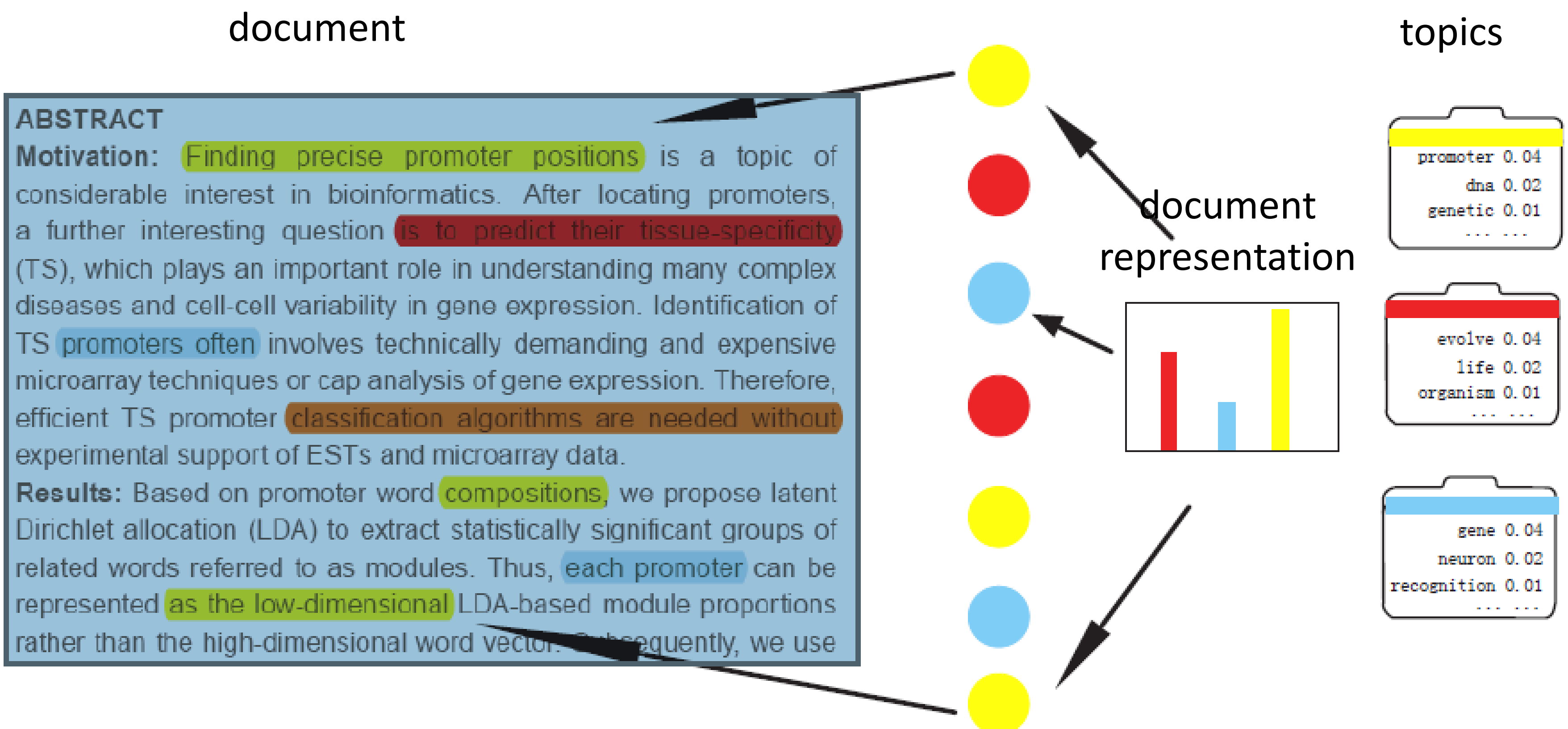
- Topic Modeling
- Methods of Matching with Topic Model
- Two Approaches to Topic Modeling

Topic Modeling



- Input
  - Document collection
- Processing
  - Discover latent topics in document collection
- Output
  - Latent topics in document collection
  - Topic representations of documents

Topics and Document Representations



Deal with Term Mismatch with Topic Model

- Topics of query and document are identified
- Match query and document through topics, although query and document do not share terms

Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic7	Topic8	Topic9	Topic10
OPEC	Africa	contra	school	Noriega	firefight	plane	Saturday	Iran	senate
oil	South	Sandinista	student	Panama	ACR	crash	coastal	Iranian	Reagan
cent	African	rebel	teacher	Panamanian	forest	flight	estimate	Iraq	billion
barrel	Angola	Nicaragua	education	Delval	park	air	western	hostage	budget
price	apartheid	Nicaraguan	college	canal	blaze	airline	Minsch	Iraqi	Trade

Methods of Matching Using Topic Model

- Topic level matching
  - Probabilistic model: PLSI (Hofmann ’99), LDA (Blei et al., ’03)
  - Non-probabilistic model: LSI (Deerwester et al., ’88), NMF (Lee & Seung ’00), RLSI (Wang et al., ’11), GMF (Wang et al., ’12)
- Document smoothing
  - Clustering-based (Kurland & Lee ’04, Diaz ’05)
  - LDA-based (Wei & Croft ’06)
- Query smoothing
  - PLSI-based (Yi & Allan ’09)

Topic Level Matching

- Representing query and document as topic distributions (or topic vectors)
  - $\mathbf{q} \rightarrow P(z|\mathbf{q})$
  - $\mathbf{d} \rightarrow P(z|\mathbf{d})$
- Similarities
  - Cosine similarity
  - Symmetric KL-divergence:
    $\sum_z \left(P(z|\mathbf{q}) \ln \frac{P(z|\mathbf{q})}{P(z|\mathbf{d})} \right) + \sum_z \left(P(z|\mathbf{d}) \ln \frac{P(z|\mathbf{d})}{P(z|\mathbf{q})} \right)$

  …

Representing query/doc with topics

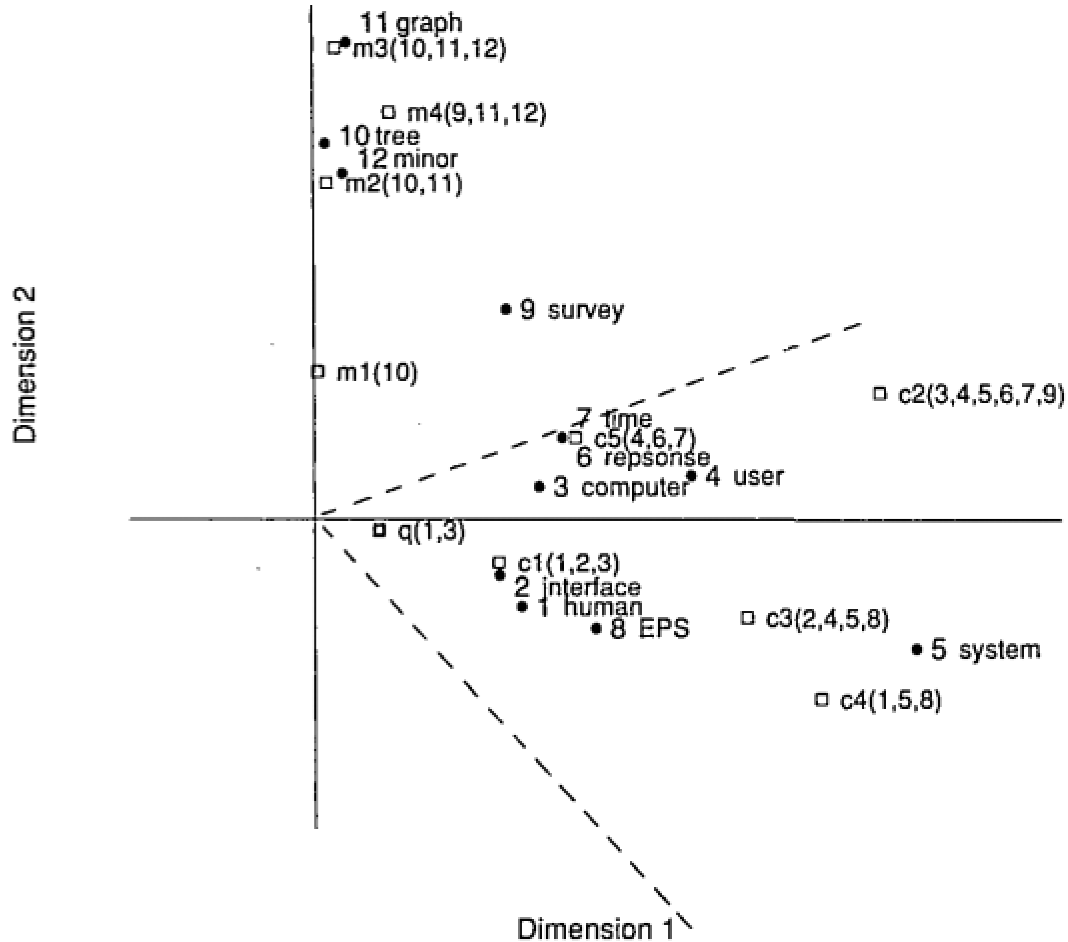


FIG. 1. A two-dimensional plot of 12 Terms and 9 Documents from the sampe TM set. Terms are represented by filled circles. Documents are shown as open squares, and component terms are indicated parenthetically. The query ("human computer interaction") is represented as a pseudo-document at point q. Axes are scaled for Document-Document or Term-Term comparisons. The dotted cone represents the region whose points are within a cosine of .9 from the query q. All documents about human-computer (c1–c5) are "near" the query (i.e., within this cone), but none of the graph theory documents (m1–m4) are nearby. In this reduced space, even documents c3 and c5 which share no terms with the query are near it.

Document Smoothing with Topics (Wei & Croft, 2006)

- Topic model: PLSI

$$P_{PLSI}(w|\mathbf{d}) = \sum_{z} P(w|z)P_{PLSI}(z|\mathbf{d})$$

- Topic model: LDA

$$P_{LDA}(w|\mathbf{d}) = \sum_{z} P(w|z)P_{LDA}(z|\mathbf{d})$$

- Combination of language model and topic model

$$P(w|\mathbf{d}) = \alpha P_{LM}(w|\mathbf{d}) + (1-\alpha)P_{TM}(w|\mathbf{d})$$

Query Smoothing with Topic Model (Yi & Allan, 2009)

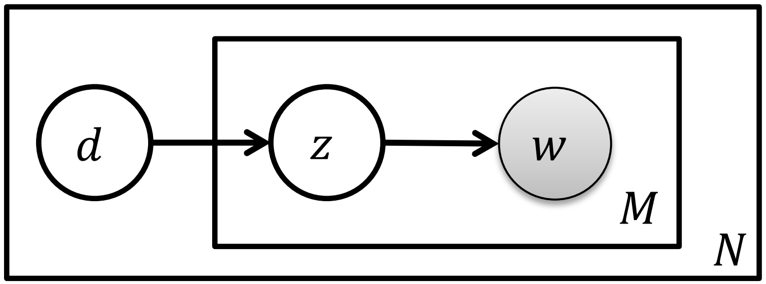
- Topic model

$$P_{TM}(w|\mathbf{q}) = \sum\nolimits_{z} P(w|z)P(z|\mathbf{q})$$

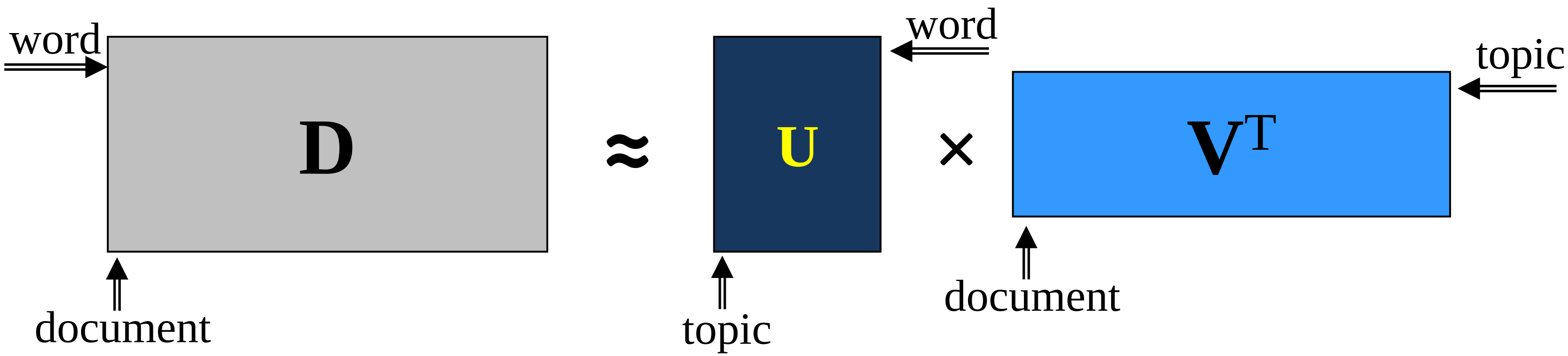
- Generate words from topic model
- Query expansion with generated words

Topic Modeling: Two Approaches

- Probabilistic approach



- Non-probabilistic approach



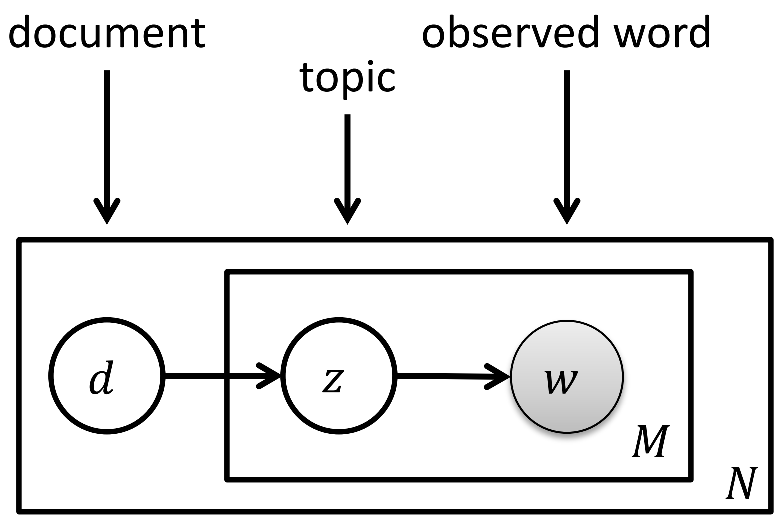
Topic Modeling: Two Approaches (cont’)

- Probabilistic approach
  - Model: probabilistic model (graphical model)
  - Learning: maximum likelihood estimation
  - Methods: PLSI, LDA
- Non-probabilistic approach
  - Model: vector space model
  - Learning: matrix factorization
  - Methods: LSI, NMF, RLSI
- Non-probabilistic models can be reformulated as probabilistic models

Probabilistic Topic Model

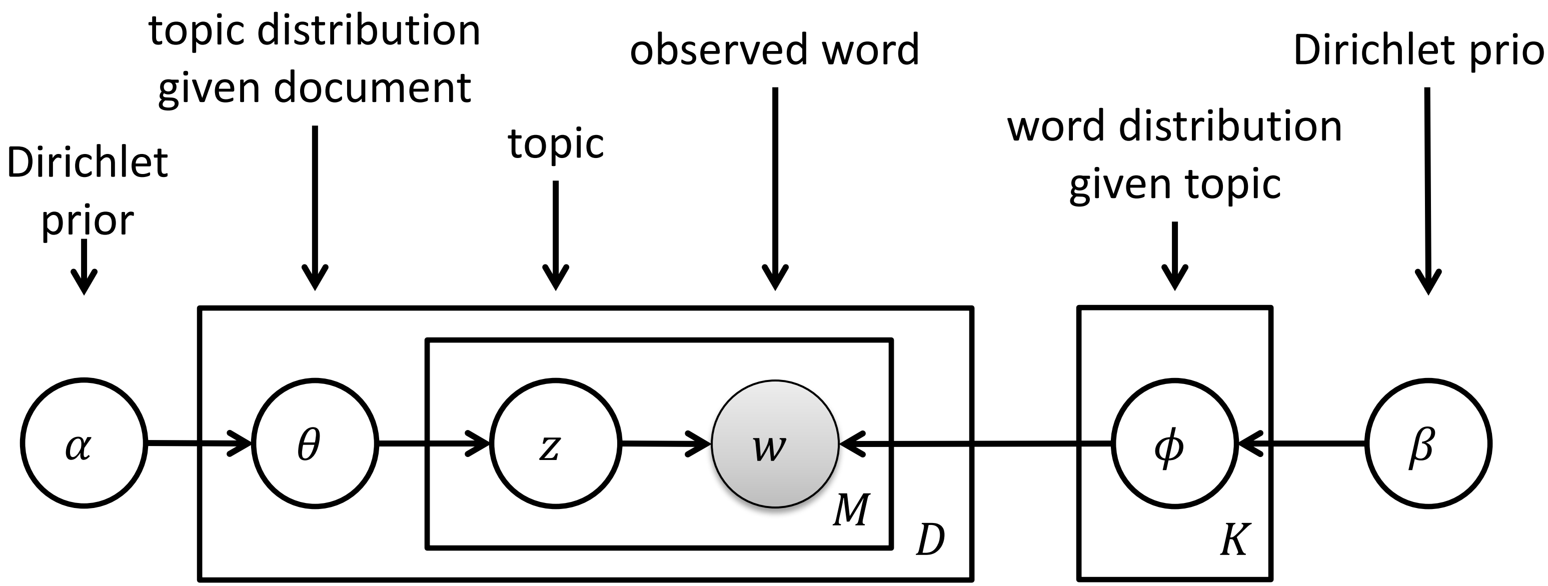
- Topic: probability distribution over words
- Document: probability distribution over topics
- Graphical model
  - Word, topic, document, and topic distribution are represented as nodes
  - Probabilistic dependencies are represented as directed edges
  - Generation process
- Interpretation: soft clustering

Probabilistic Latent Semantic Indexing (Hofmann 1999)



- For each document
  - Generate doc d with probability $P(d)$
  - For each word
    - Generate topic z with probability $P(z|d)$
    - Generate word w with probability $P(w|z)$

Latent Dirichlet Allocation (Blei et al., 2003)



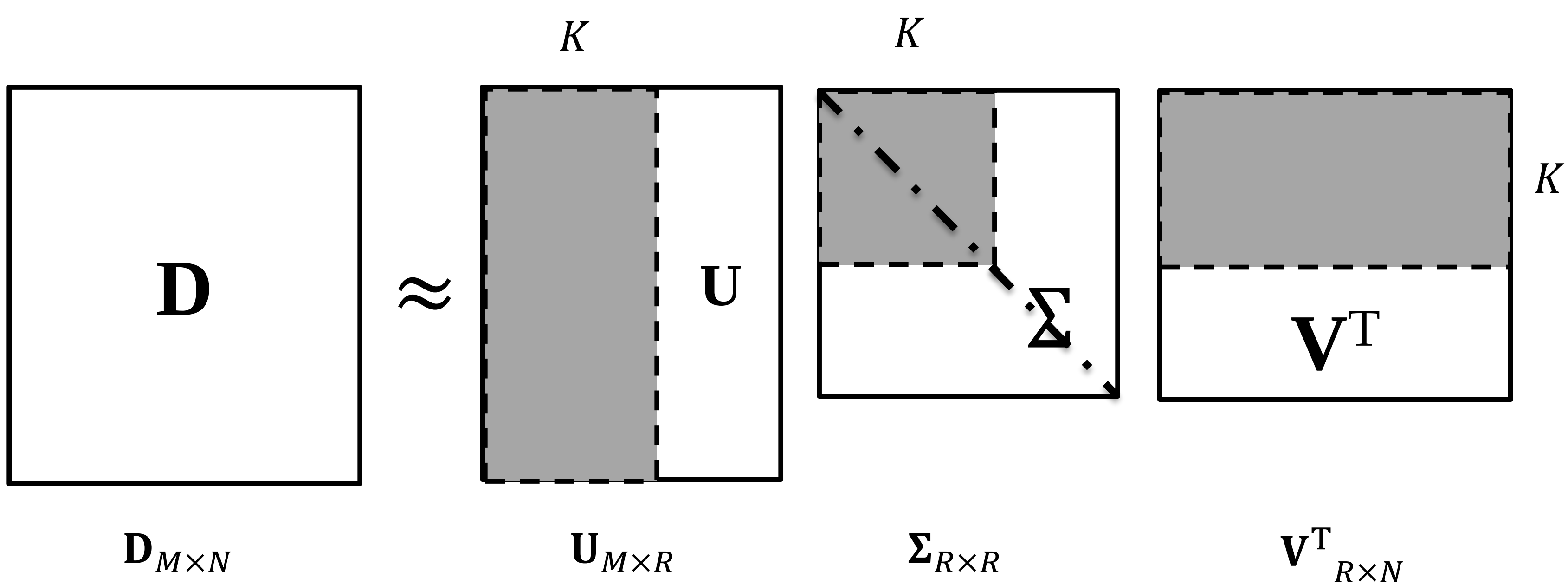
- Generation process
  - Word distribution given topic $\phi \sim \mathrm{Dir}(\beta)$
  - For each document:
    - Determine topic distribution $\theta \sim \mathrm{Dir}(\alpha)$
    - For each word:
      - Generate topic $z \sim \mathrm{Mul}(\theta)$
      - Generate word $w \sim \mathrm{Mul}(\phi)$

Non-probabilistic Topic Model

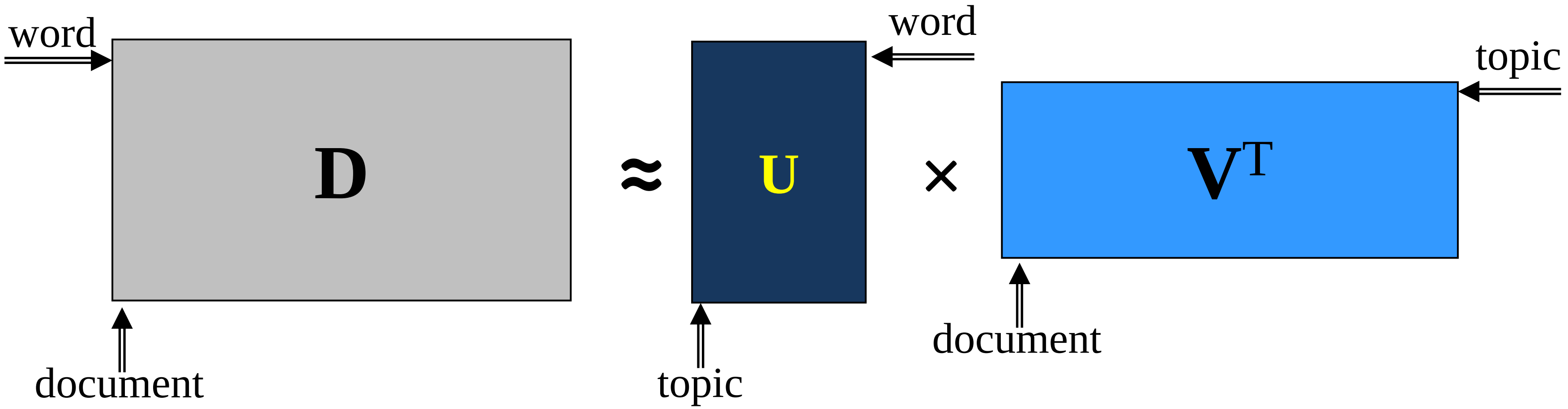
- Document: vector of words
- Topic: vector of words
- Document representation: combination of topic vectors
- Matrix factorization
- Interpretation: projection to topic space

Latent Semantic Indexing (Deerwester et al., 1990)

- Representing document collection with co-occurrence matrix (TF or TFIDF)
- Performing Singular Value Decomposition (SVD) and producing k-dimensional topic space



Nonnegative Matrix Factorization (Lee and Seung, 2001)

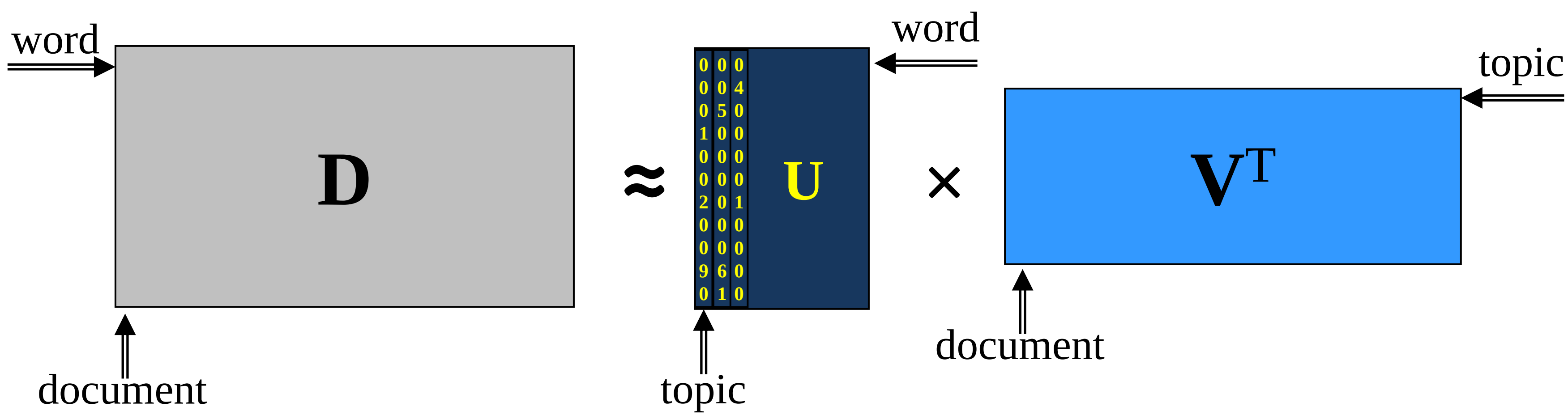


- **U** and **V** are nonnegative

$$\min_{\mathbf{U},\mathbf{V}} \left\| \mathbf{D} - \mathbf{U}\mathbf{V}^{\mathrm{T}} \right\|_F$$

$$s.t.\, u_{ij} \geq 0; v_{ij} \geq 0$$

Regularized Latent Semantic Indexing (Wang et al., 2011)



- Topics are sparse

word representation of doc n

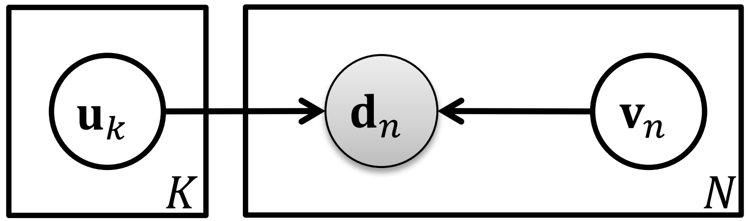
topic matrix

topic representation of doc n

$$\min_{\mathbf{U},\mathbf{V}} \sum_{n=1}^{N} \|\mathbf{d}_n - \mathbf{U}\mathbf{v}_n\|_2^2 + \lambda_1 \sum_{k=1}^{K} \|\mathbf{u}_k\|_1 + \lambda_2 \sum_{n=1}^{N} \|\mathbf{v}_n\|_2^2$$

topics are sparse

Probabilistic Interpretation of Nonprobabilistic Models (RLSI)



$$\min_{\mathbf{U},\mathbf{V}} \sum_{n=1}^{N} \|\mathbf{d}_n - \mathbf{U}\mathbf{v}_n\|_2^2 + \lambda_1 \sum_{k=1}^{K} \|\mathbf{u}_k\|_1 + \lambda_2 \sum_{n=1}^{N} \|\mathbf{v}_n\|_2^2$$

- Document generated according to Gaussian distribution

$$P(\mathbf{d}_n|\mathbf{U},\mathbf{v}_n) \propto \exp(-\|\mathbf{d}_n - \mathbf{U}\mathbf{v}_n\|_2^2)$$

- Laplacian prior

$$P(\mathbf{u}_k) \propto \exp(-\lambda_1 \|\mathbf{u}_k\|_1)$$

- Gaussian prior

$$P(\mathbf{v}_n) \propto \exp(-\lambda_2 \|\mathbf{v}_n\|_2^2)$$

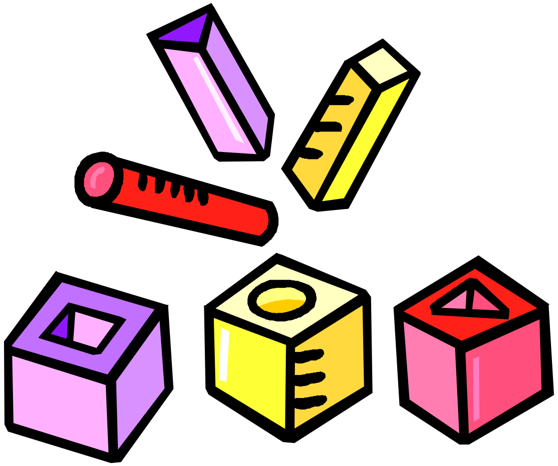
References

- David Andrzejewski and David Buttler. Latent Topic Feedback for Information Retrieval. In Proc. of KDD 2011.
- Paul N. Bennett, Krysta Svore, and Susan T. Dumais. Classification Enhanced Ranking. In Proc. of WWW 2010.
- David Blei, Andrew Ng, Michael Jordan, John Lafferty. Latent Dirichlet allocation. JMLR, 2003.
- Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman. Indexing by Latent Semantic Analysis, JASIST, 1990.
- Fernando Diaz. Regularizing ad hoc retrieval scores. In Proc. of CIKM 2005.
- Martin Franz and Jeffery S McCarley. Information retrieval with non-negative matrix factorization. IBM Patent. 2001.
- Thomas Hofmann, Probabilistic Latent Semantic Indexing. In Proc. of SIGIR 1999.
- Oren Kurland and Lillian Lee. Corpus Structure, Language Models, and Ad Hoc Information Retrieval. In Proc. of SIGIR 2004.
- Daniel D. Lee and H. Sebastian Seung. Algorithms for Non-negative Matrix Factorization. In Proc. of NIPS 2000.
- Michael L. Littman, Susan T. Dumais, and Thomas K. Landauer. Automatic Cross-Language Information Retrieval using Latent Semantic Indexing. Cross-Language Information Retrieval, 1996.

References

- Xiaoyong Liu and W. Bruce Croft. Cluster-based Retrieval using Language models. In Proc. of SIGIR 2004.
- David Mimno, Hanna Wallach, Jason Naradowsky, David Smith, and Andrew McCallum. Ploylingual Topic models. In Proc. of EMNLP 2009
- Quan Wang, Jun Xu, Hang Li, and Nick Craswell. Regularized Latent Semantic Indexing. In Proc. of SIGIR 2011.
- Xing Wei and W. Bruce Croft. LDA-based Document Models for Ad-hoc Retieval. In Proc. of SIGIR 2006.
- Jinxi Xu and W. Bruce Croft. Cluster-based Language Models for Distributed Retrieval. In Proc. of SIGIR 1999.
- Xing Yi and James Allan. A comparative study of utilizing topic models for Information Retrieval. In Proc. of ECIR 2009.

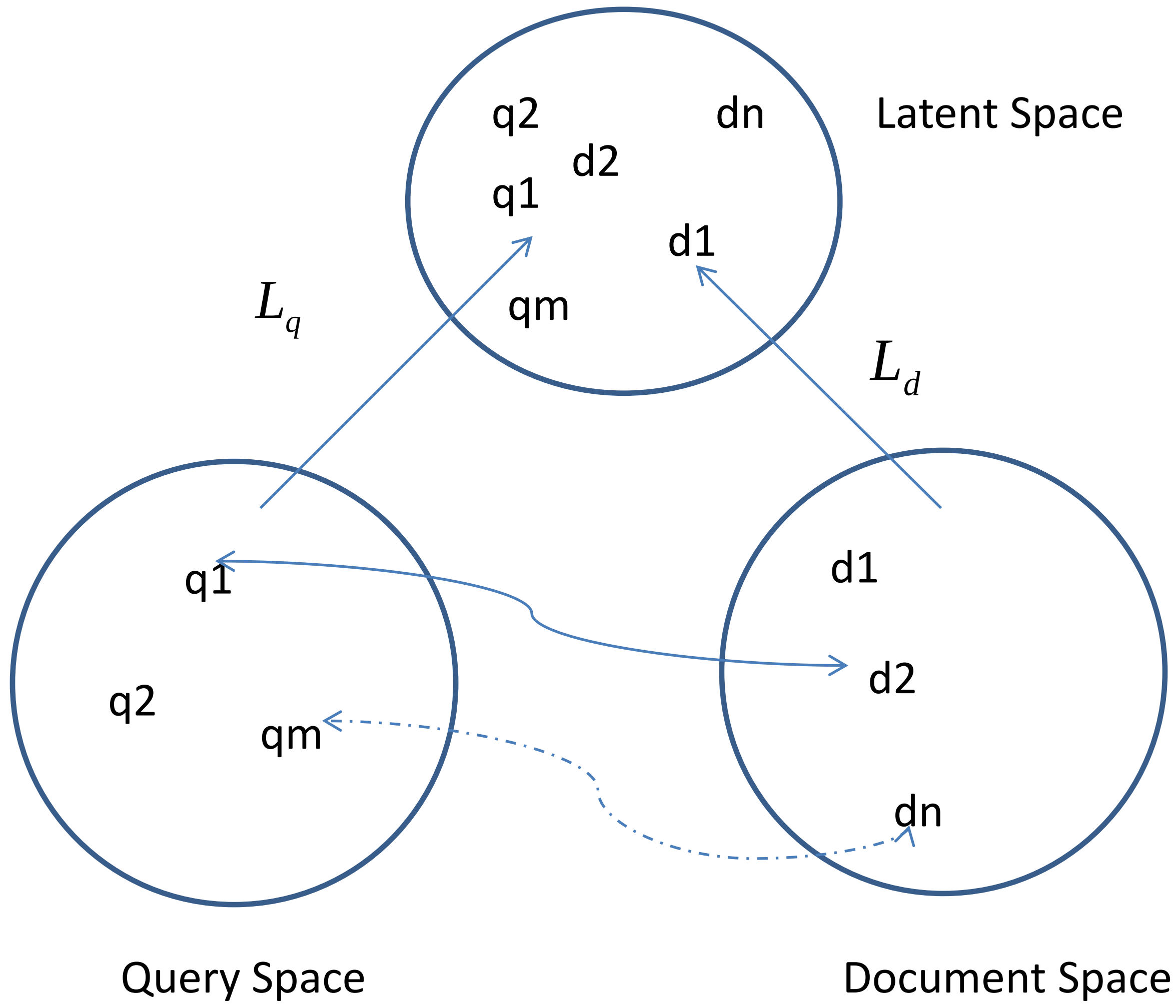
6. Matching in Latent Space



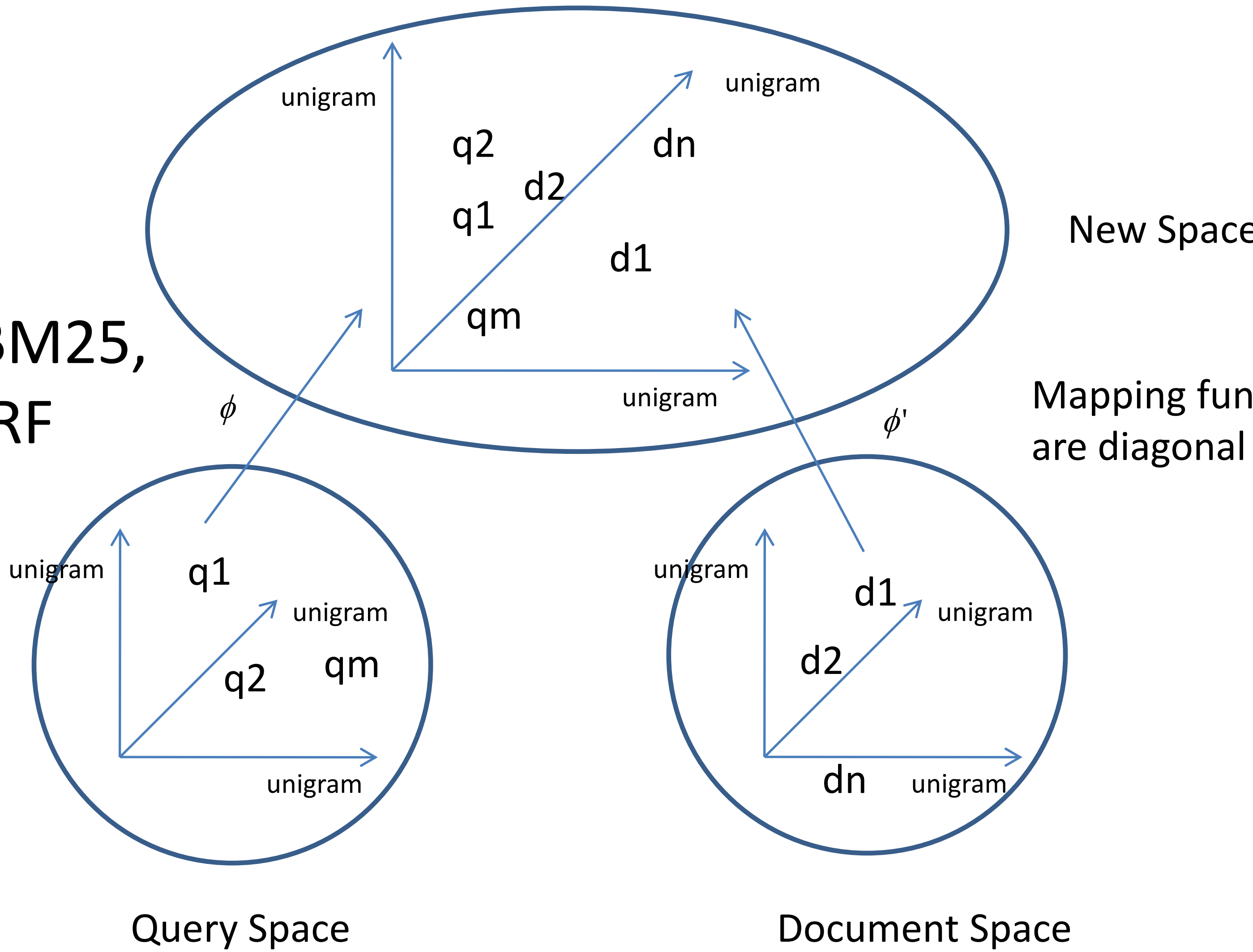
Matching in Latent Space

- Motivation
  - Matching between query and document in latent space
- Assumption
  - Queries have similarity
  - Document have similarity
  - Click-through data represent “similarity” relations between queries and documents
- Approach
  - Projection to latent space
  - Regularization or constraints
- Results
  - Significantly enhance accuracy of query document matching

Matching in Latent Space



IR Models as Similarity Functions (Xu and Li 2010)



VSM, BM25, LM, MRF

Mapping functions are diagonal matrices

IR Models Are Similarity Functions

- VSM
  - $\mathrm{BM25}(q,d) = \left\langle \phi_Q^{VSM}(q), \phi_D^{VSM}(d) \right\rangle$, for all $w \in V$
    $\phi_Q^{VSM}(q)_w = tfidf(w,q)$ and $\phi_D^{VSM}(d)_w = tfidf(w,d)$

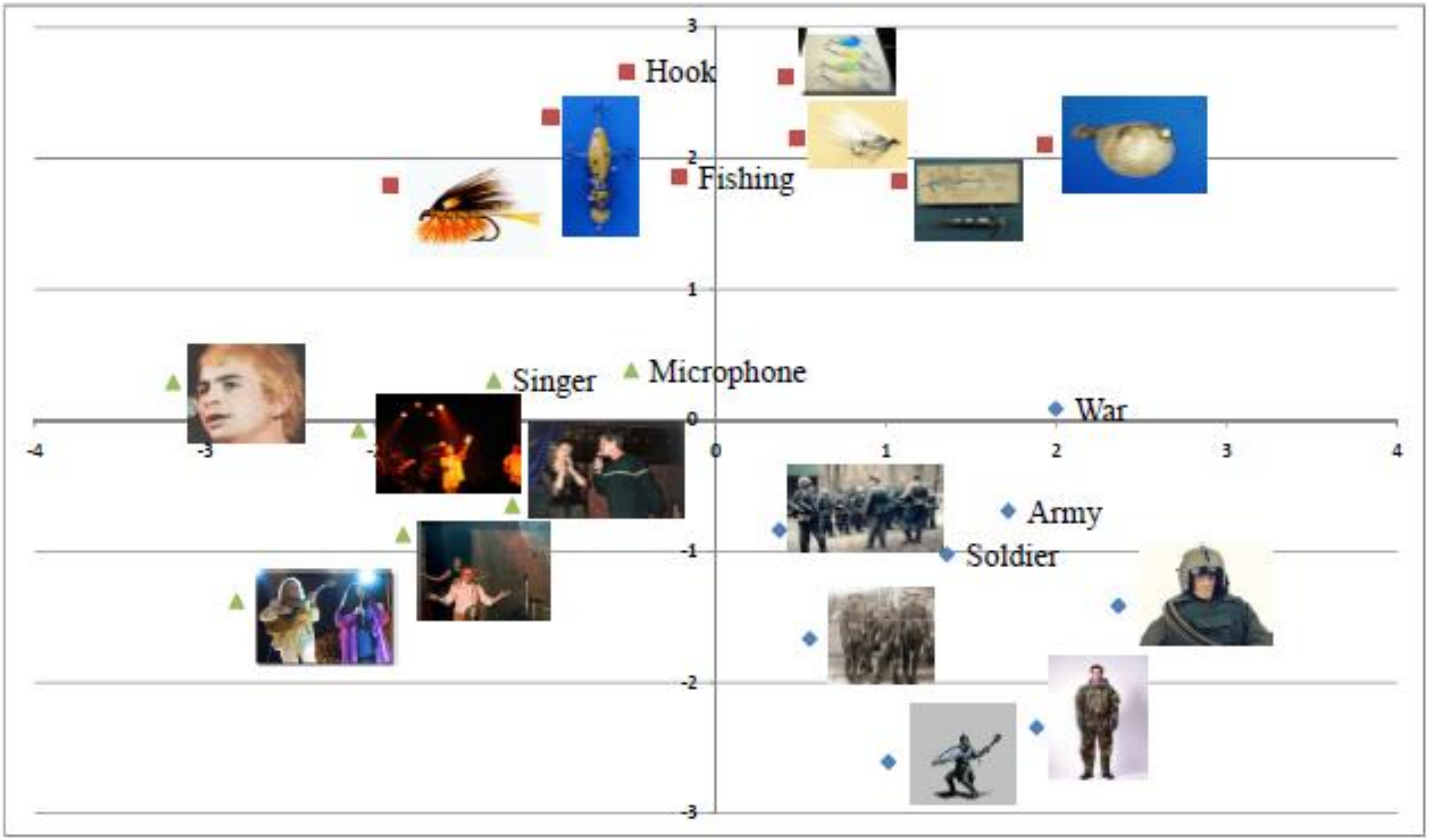
- BM25
  - $\mathrm{BM25}(q,d) = \left\langle \phi_Q^{BM25}(q), \phi_D^{BM25}(d) \right\rangle$, for all $w \in V$
    $\phi_Q^{BM25}(q)_w = \frac{(k_3+1)\times tf(w,q)}{k_3+tf(w,q)}$
    $\phi_D^{BM25}(d)_w = \mathrm{IDF}(w) \cdot \frac{(k_1+1)\times tf(w,d)}{k_1\left(1-b+b\cdot\frac{len(d)}{avgDocLen}\right)+tf(w,d)}$

- LMIR
  - $\mathrm{LMIR}(q,d) = \left\langle \phi_Q^{LMIR}(q), \phi_D^{LMIR}(d) \right\rangle + len(q) \cdot \log\frac{\mu}{len(d)+\mu}$, for all $w \in V$
    $\phi_Q^{LMIR}(q)_w = tf(w,q)$
    $\phi_D^{LMIR}(d)_w = \log\left(1+\frac{tf(w,d)}{\mu\cdot P(w)}\right)$, where $P(w)$ plays similar role as IDF in BM25

Problem with IR Models: Term Mismatch

- Matching in Latent Space can solve the problem by
  - Reducing dimensionality of latent space (from term level matching to semantic matching)
  - Correlating semantically similar terms (matrices are not diagonal)
  - Automatically learning mapping functions from data
- *Generalized and Learnable of IR models*

Example: Projecting Keywords and Images into Latent Space



Partial Least Square (PLS)

- Setting
  - Two spaces: $\mathcal{X} \subset \mathbb{R}^m$ and $\mathcal{Y} \subset \mathbb{R}^n$.
- Input
  - Training data: $\{(x_i, y_i, r_i)\}_{1 \le i \le N}$, $r_i \in \{+1, -1\}$ or $r_i \in R$
- Output
  - Similarity function $f(x, y)$
- Assumption
  - Two linear (and orthonormal) transformations $L_\mathcal{X}$ and $L_\mathcal{Y}$
  - Dot product as similarity function $\langle {L_\mathcal{X}}^T x, \ {L_\mathcal{Y}}^T y \rangle = x^T L_\mathcal{X} {L_\mathcal{Y}}^T y$
- Optimization

$$argmax_{L_\mathcal{X}, L_\mathcal{Y}} \sum_{r_i=+1} {x_i}^T L_\mathcal{X} {L_\mathcal{Y}}^T y_i - \sum_{r_i=-1} {x_i}^T L_\mathcal{X} {L_\mathcal{Y}}^T y_i$$

$$subject\ to\ {L_\mathcal{X}}^T L_\mathcal{X} = I_{k \times k}, {L_\mathcal{Y}}^T L_\mathcal{Y} = I_{k \times k}$$

Solution of Partial Least Square

- Non-convex optimization
- Can prove that global optimal solution exists
- Global optimal can be found by solving SVD (Singular Value Decomposition)
- SVD of Matrix $M_s - M_D = U\Sigma V^T$

Regularized Mapping to Latent Space (RMLS)

- Setting
  - Two spaces: $\mathcal{X} \subset \mathbb{R}^m$ and $\mathcal{Y} \subset \mathbb{R}^n$.
- Input
  - Training data: $\{(x_i, y_i, r_i)\}_{1 \le i \le N}$, $r_i \in \{+1, -1\}$ or $r_i \in R$
- Output
  - Similarity function $f(x, y)$
- Assumption
  - L1 and L2 regularization on $L_{\mathcal{X}}$ and $L_{\mathcal{Y}}$ (sparse transfromations)
  - Dot product as similarity function $\langle L_{\mathcal{X}}{}^T x, \ L_{\mathcal{Y}}{}^T y \rangle = x^T L_{\mathcal{X}} L_{\mathcal{Y}}{}^T y$
- Optimization

$$argmax_{L_{\mathcal{X}}, L_{\mathcal{Y}}} \sum_{r_i=+1} x_i{}^T L_{\mathcal{X}} L_{\mathcal{Y}}{}^T y_i - \sum_{r_i=-1} x_i{}^T L_{\mathcal{X}} L_{\mathcal{Y}}{}^T y_i$$

$$subject\ to\ |lx| \le \vartheta x,\ |ly| \le \vartheta y,\ \| lx \| \le \lambda x,\ \| ly \| \le \lambda y,$$

Solution of Regularized Mapping to Latent Space

- Coordinate Descent
- Repeat
  - Fix Lx, updateLy
  - Fix Ly, updateLx
- Update can be parallelized by rows

Comparison

	PLS	RMLS
Assumption	Orthogonal	L1 and L2 Regularization
Optimization Method	Singular Value Decomposition	Coordinate Descent
Optimality	Global optimum	Local optimum
Efficiency	Low	High
Scalability	Low	High

Experimental Results

Enterprise Search

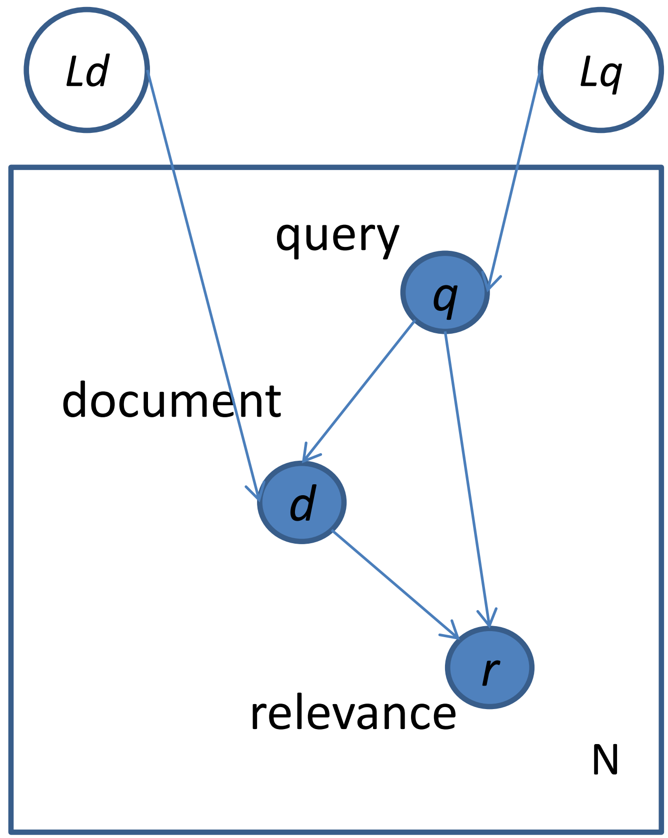
	NDCG@1	NDCG@3	NDCG@5
$MPLS_{Com}$	**0.715**	**0.733**	**0.747**
$MPLS_{Conca}$	0.700	0.728	0.742
$MPLS_{Word}$	0.688	0.718	0.739
$MPLS_{Bipar}$	0.659	0.684	0.705
BM25	0.653	0.657	0.663
RW	0.654	0.683	0.700
RW+BM25	0.664	0.688	0.705
LSI	0.656	0.676	0.695
LSI+BM25	0.692	0.701	0.712

Web Search

	NDCG@1	NDCG@3	NDCG@5
$MPLS_{Com}$	**0.681**	**0.731**	**0.739**
$MPLS_{Conca}$	0.676	0.728	0.736
$MPLS_{Word}$	0.674	0.726	0.732
$MPLS_{Bipar}$	0.612	0.680	0.693
BM25	0.637	0.690	0.690
RW	0.655	0.704	0.704
RW+BM25	0.671	0.718	0.716
LSI	0.588	0.665	0.676
LSI+BM25	0.649	0.705	0.706

- RMLS and PLS work better than BM25, Random Walk, Latent Semantic Indexing
- RMLS works equally well as PLS, with higher learning efficiency and scalability

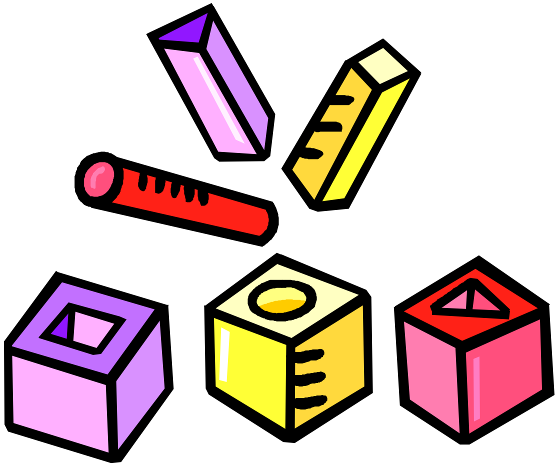
Graphical Model Representation



References

- D.R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. Neural Computation, 2004.
- Jianfeng Gao, Kristina Toutanova and Wen-tau Yih. Clickthrough-based latent semantic models for web search. In Proc. of SIGIR, 2011.
- R. Rosipal and N. Krämer. Overview and recent advances in partial least squares. Subspace, Latent Structure and Feature Selection, 2006.
- Wei Wu, Hang Li, and Jun Xu. Learning Query and Document Similarities from Click-through Bipartite Graph with Metadata. Microsoft Research Technical Report, 2011.
- Jun Xu, Hang Li, Chaoliang Zhong, Relevance Ranking Using Kernels, In Proceedings of the 6th Asian Information Retrieval Societies Symposium (AIRS'10), Best Paper Award, 1-12, 2010.

7. Generalization: Learning to Match

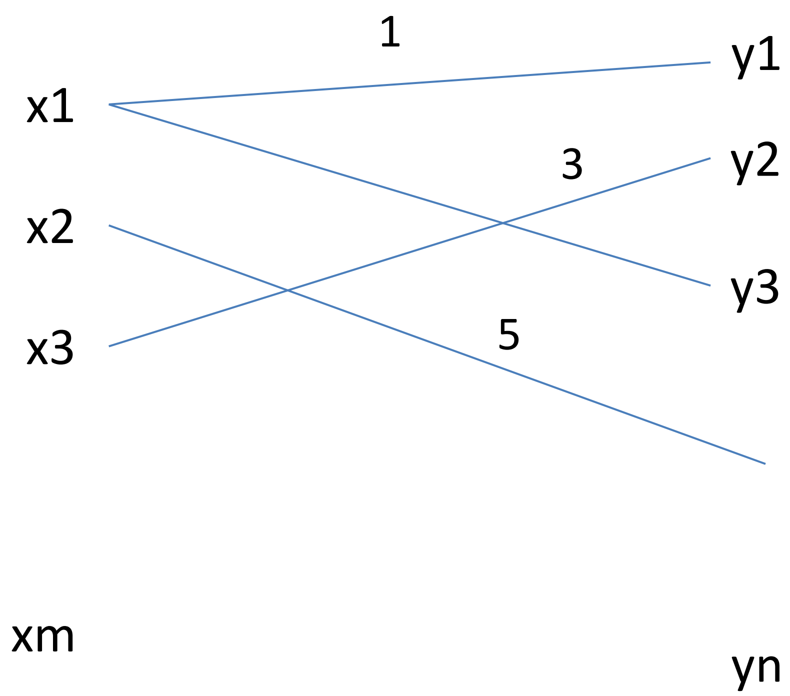


Matching between Heterogeneous Data is Everywhere

- Matching between user and product (collaborative filtering)
- Matching between text and image (image annotation)
- Matching between people (dating)
- Matching between languages (machine translation)
- Matching between receptor and ligand (drug design)

Matching Problem: Instance Matching

Graph View

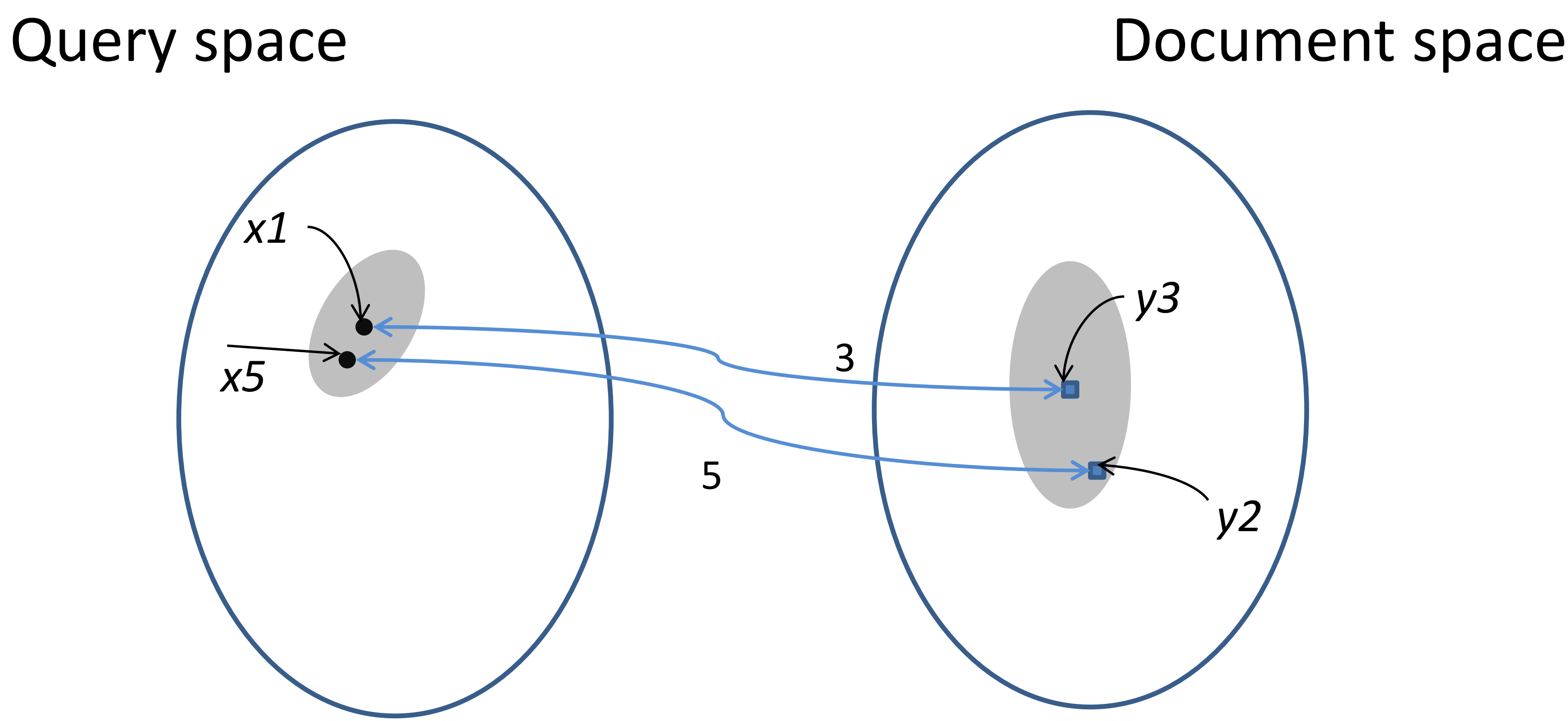


Matching Problem: Instance Matching

Matrix View

	y1	y2	y3			yn
x1			1			
x2						1
x3				4		
		1			5	
xm						

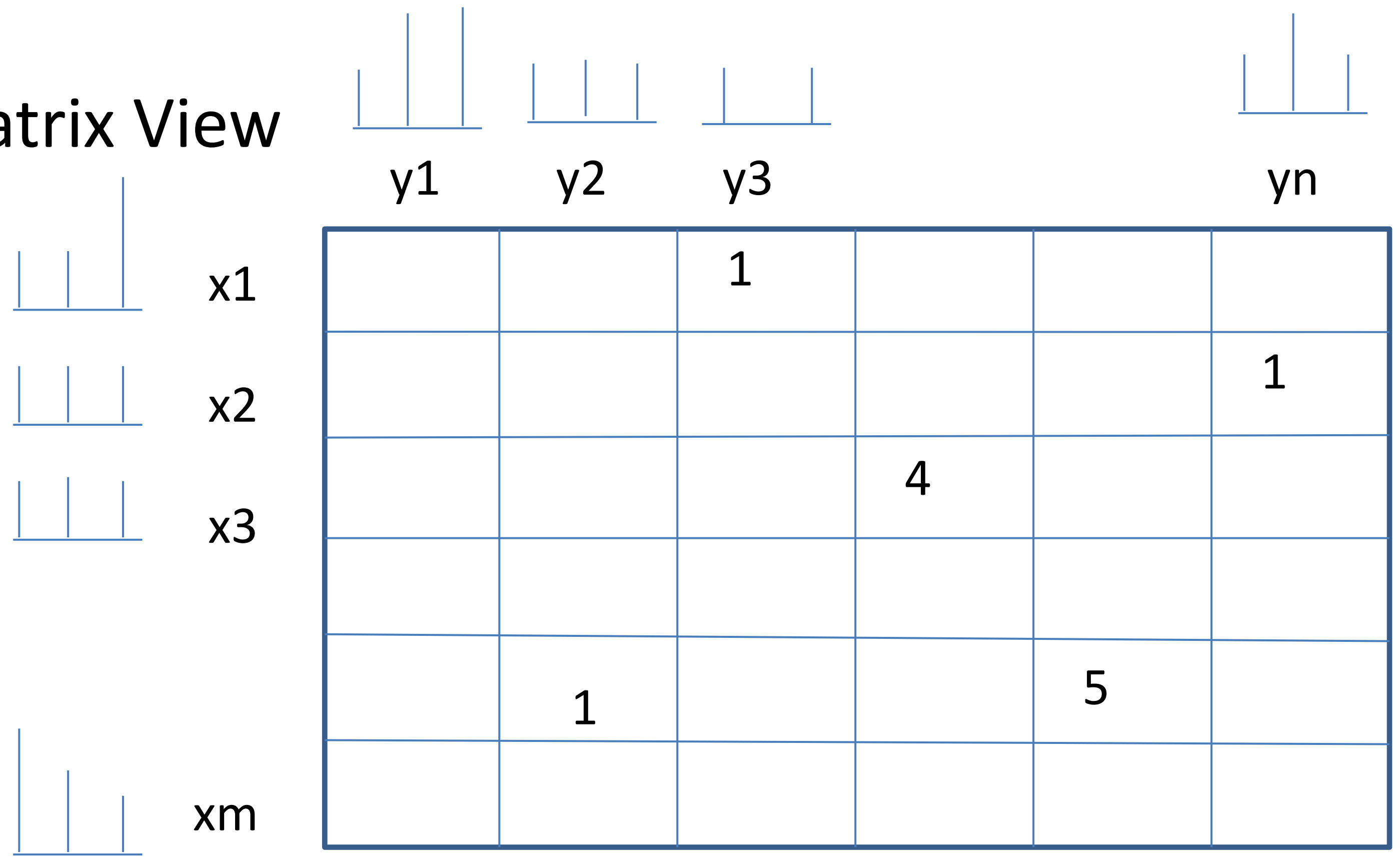
Matching Problem: Content Matching



Space View

Matching Problem: Content Matching

Matrix View



Formulation of Learning Problem

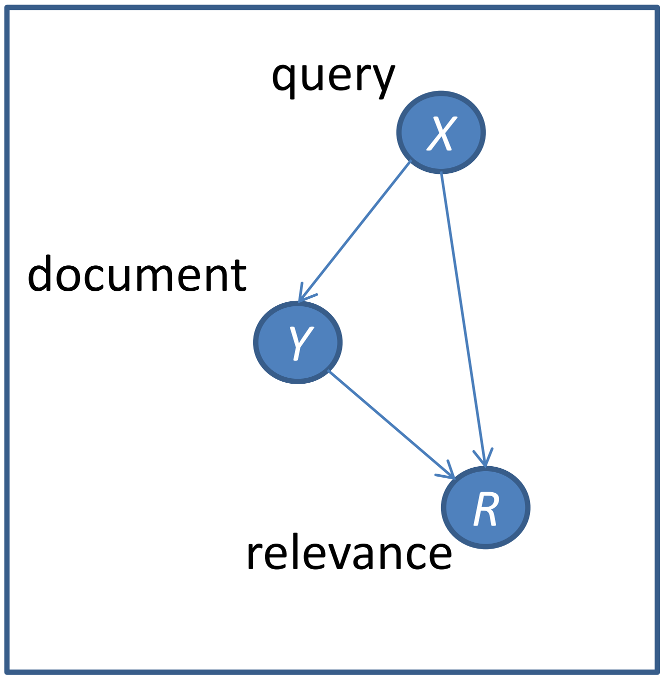
- Learning matching function

$$f(x, y)$$

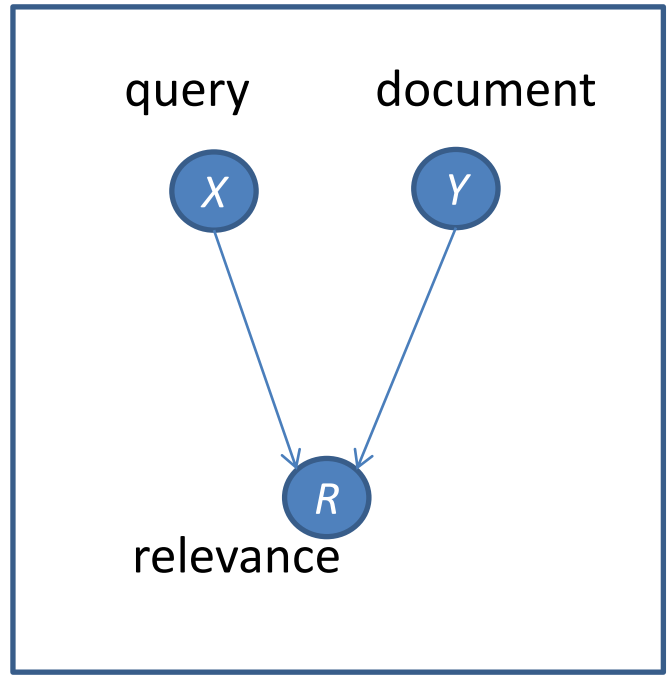
- Training data $(x_1, y_1, r_1), \cdots, (x_N, y_N, r_N)$
- Generated according to

$$x \sim P(X), \quad y \sim P(Y \mid X), \quad r \sim P(R \mid X, Y)$$

Graphical Model of Data Generation Process



This process



Not this process!

Formulation of Learning Problem

- Loss Function

$$L(r, f(x, y))$$

- Risk Function

$$R(r, f(x, y)) = \int_{X \times Y \times R} P(x, y, r) L(r, f(x, y)) dP(x, y, r)$$

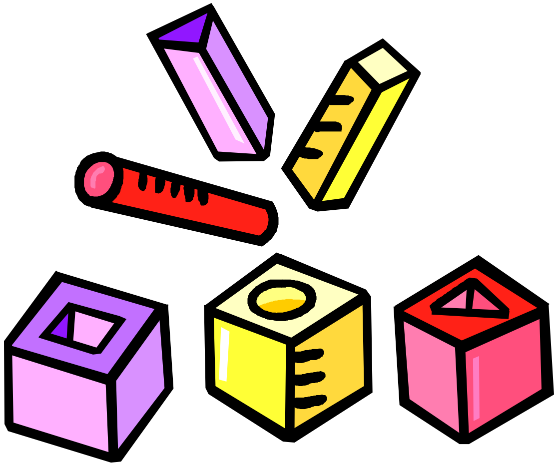
- Objective Function in Learning

$$\min_{f \in F} \sum_{i=1}^{N} L(r_i, f(x_i, y_i)) + \Omega(f)$$

References

- J. Abernethy, F. Bach, T. Evgeniou, and J.P. Vert. A new approach to collaborative filtering: Operator estimation with spectral regularization. JMLR, 2009.
- D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. IEEE PAMI, 2008.
- D.R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. Neural Computation, 2004.
- D.R. Hardoon and J. Shawe-Taylor. KCCA for different level precision in content-based image retrieval. In Proc. of Workshop on Content-Based Multimedia Indexing, 2003.
- R. Rosipal and N. Krämer. Overview and recent advances in partial least squares. Subspace, Latent Structure and Feature Selection, 2006.
- W.R. Schwartz, A. Kembhavi, D. Harwood, and L.S. Davis. Human detection using partial least squares analysis. In Proc. of ICCV, 2009.
- H. Saigo, N. Krämer, and K. Tsuda. Partial least squares regression for graph mining. In Proc. of KDD, 2008.

8. Summary and Open Problems



Summary of Tutorial

- Query document matching is biggest challenge in search
- Machine learning for matching between query and document is making progress
- Matching at term, phrase, sense, topic, and structure levels
- Matching through query, document, query-transformations
- General problem: learning to match

Approaches to Learning for Matching Between Query and Document

- Matching with Dependency Model
- Matching by Query Reformulation
- Matching with Translation Model
- Matching with Topic Model
- Matching in Latent Space

Challenges and Open Problems

- Evaluation measures
  - Cranefield approach has limitation
- Topic drift
  - Language is synonymous and polysemous
- Scalability
  - E.g., topic modeling needs large scale computing environment
- Missing information
  - Long tail challenge

Challenges and Open Problems (2)

- Divide and conquer
  - Classifying queries and building different matching models
- Existing knowledge
  - How to incorporate existing knowledge such as Wikipedia
- Natural language
  - E.g., “distance between x and y” vs “how far is x from y”
  - More natural language techniques

Thank You!

hangli.hl@huawei.com

junxu@microsoft.com