CCF ADL 2011
Beijing
Aug. 27, 2011

Learning to Match

Hang Li
Microsoft Research Asia

Talk Outline

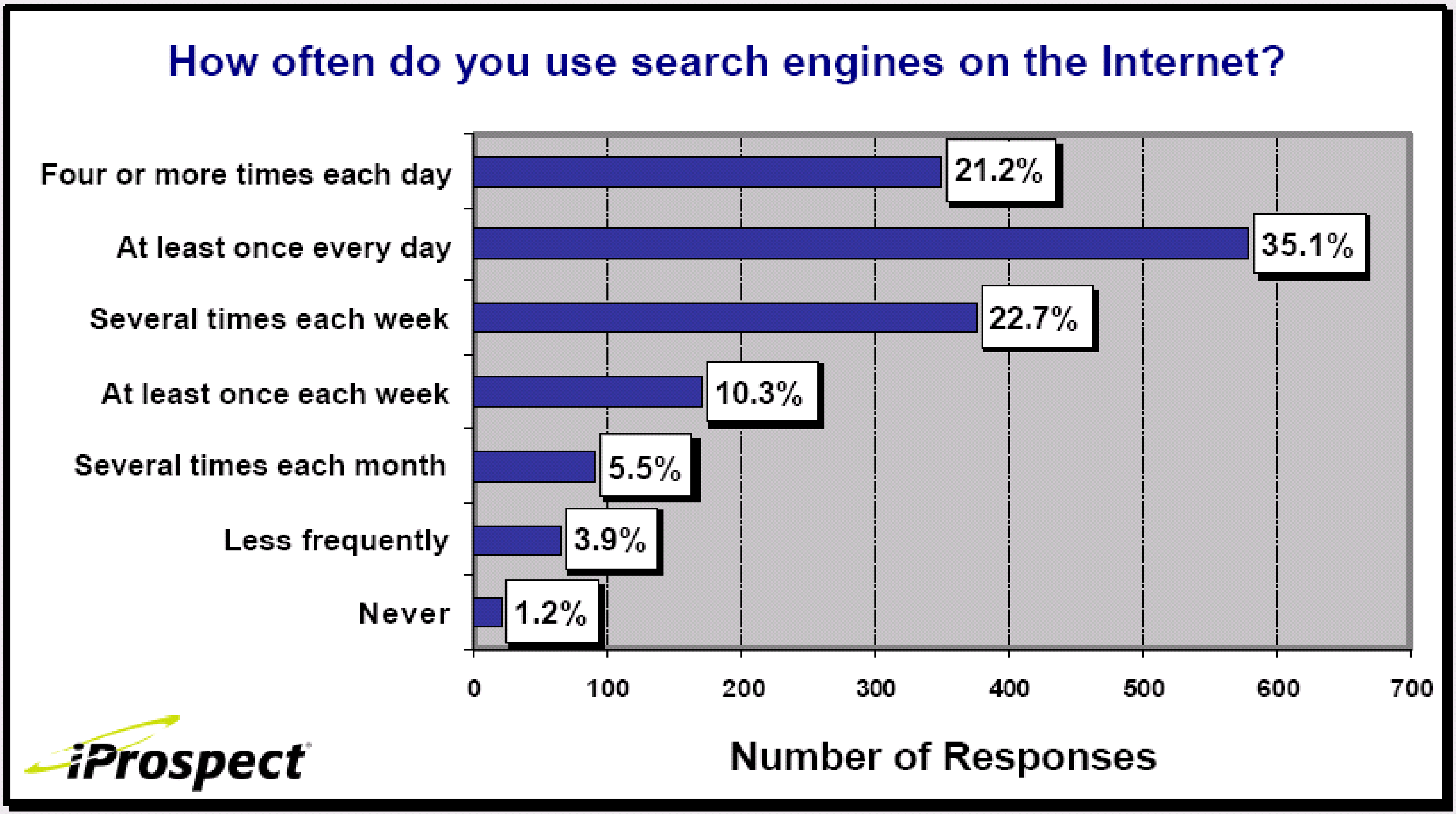
- Introduction to Web Search
- Relevance Model (Matching Model)
- Query Term Mismatch
- Learning to Match
- Our Methods
  - Robust Similarity Function Learning Using Kernel Methods
  - Regularized Latent Semantic Indexing
  - Query Generation Using Log Linear Model
  - Query Rewriting Using Conditional Random Fields

Introduction to Web Search

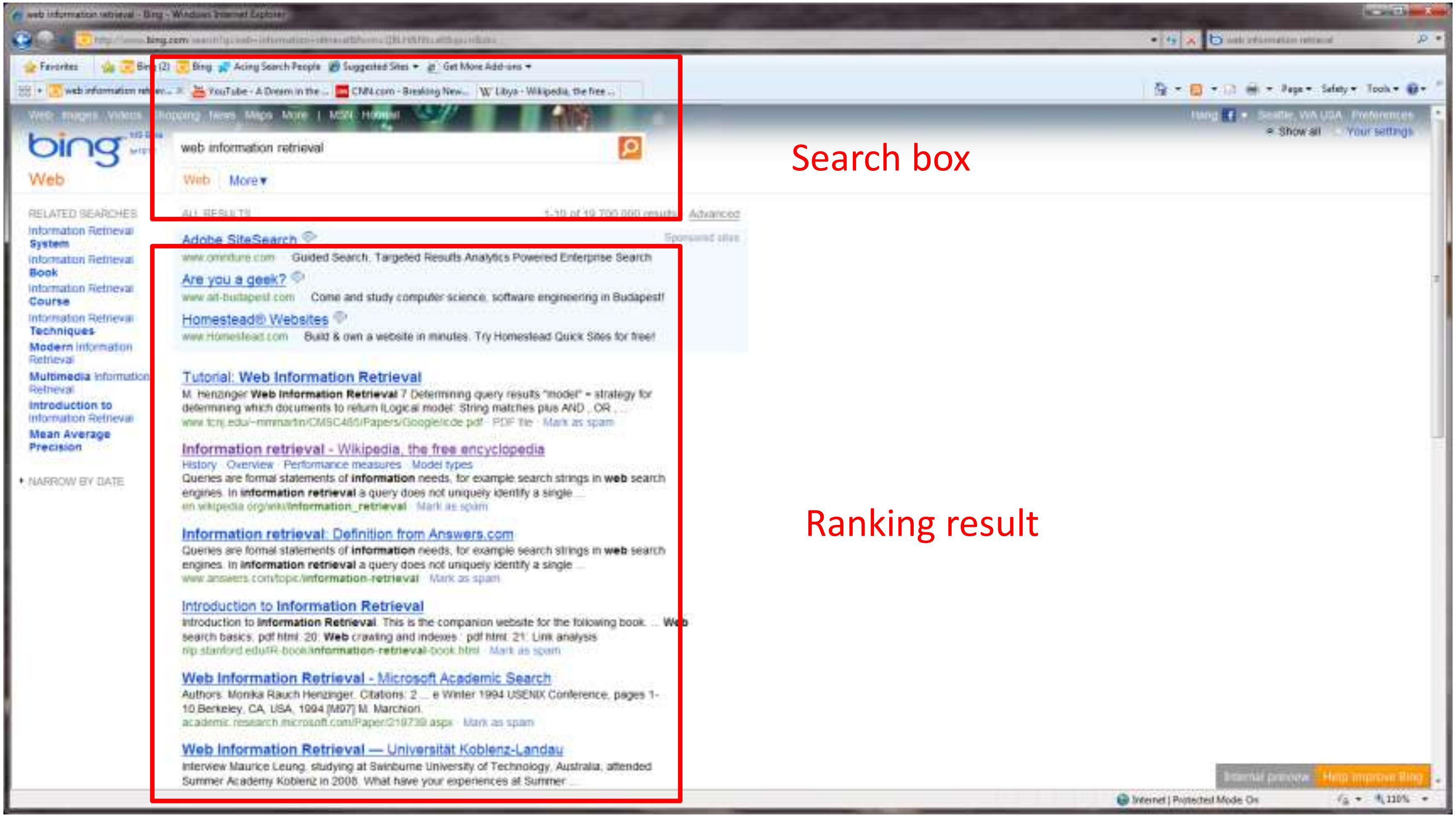
Web Search is Part of Our Life

Web Users Heavily Rely on Search Engines

http://www.iprospect.com/premiumPDFs/iProspectSurveyComplete.pdf



Simple UI



Search box
Ranking result
bing
Web
web information retrieval
RELATED SEARCHES
NARROW BY DATE
Adobe SiteSearch
Are you a geek?
Homestead® Websites
Tutorial: Web Information Retrieval
Information retrieval - Wikipedia, the free encyclopedia
Information retrieval: Definition from Answers.com
Introduction to Information Retrieval
Web Information Retrieval - Microsoft Academic Search
Web Information Retrieval — Universität Koblenz-Landau

Huge Data Center

Goal of Web Search

Effectiveness	Efficient	Easy to Use
Results are relevant	Response time is short	Good presentation
Results are comprehensive	Results are novel	Friendly user interface

Overview of Web Search Technologies

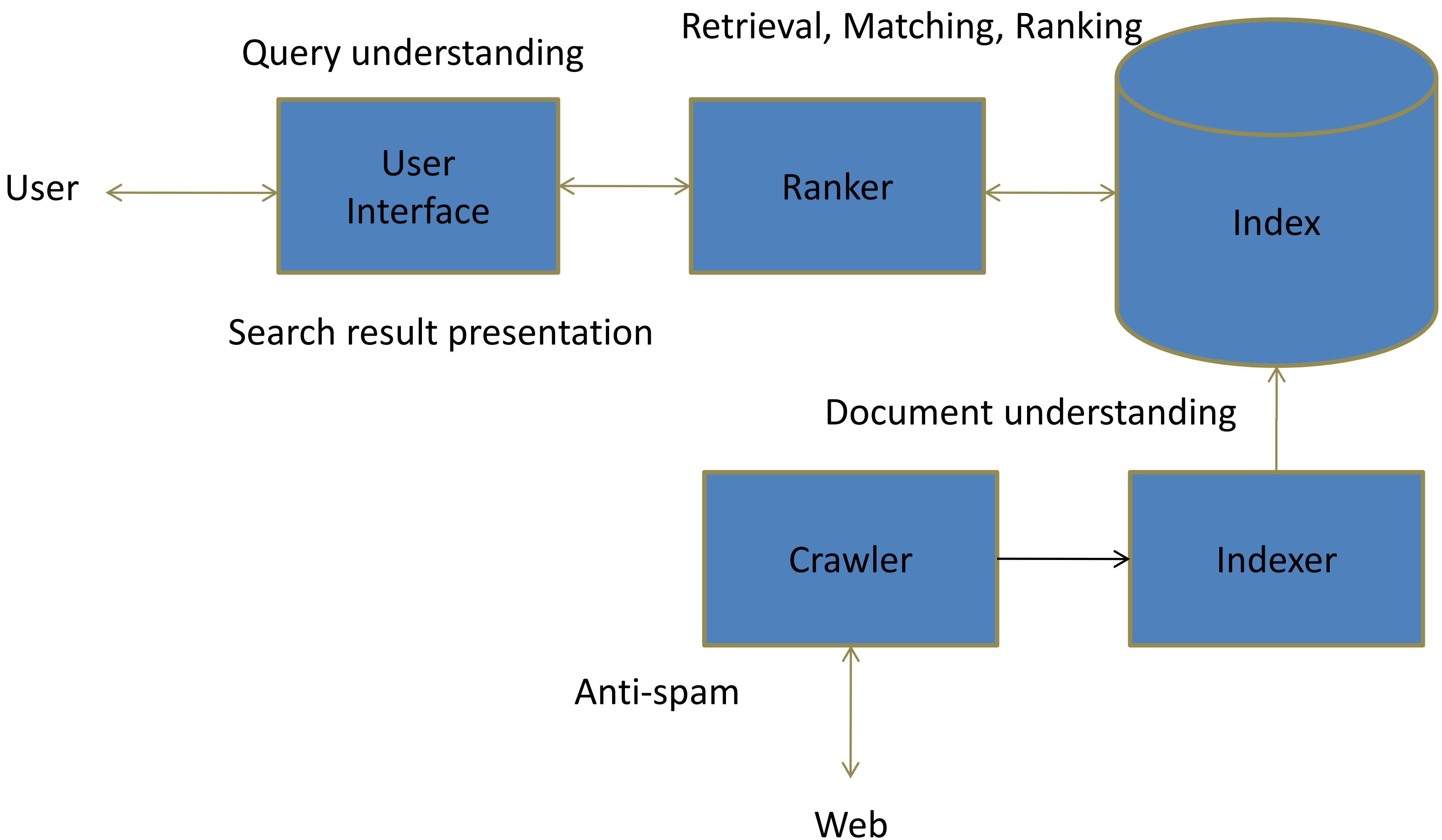
Overview of Web Search Technologies

General Web Search, Entity Search, Facet Search,
Question Answering, Image Search

Ranking, Matching, Retrieval
Document Understanding, Query Understanding,
Crawling, Indexing, Result Presentation,
Anti-Spam

Classification, Clustering, Ranking,
Graph Learning, Tagging, Distributed Computing

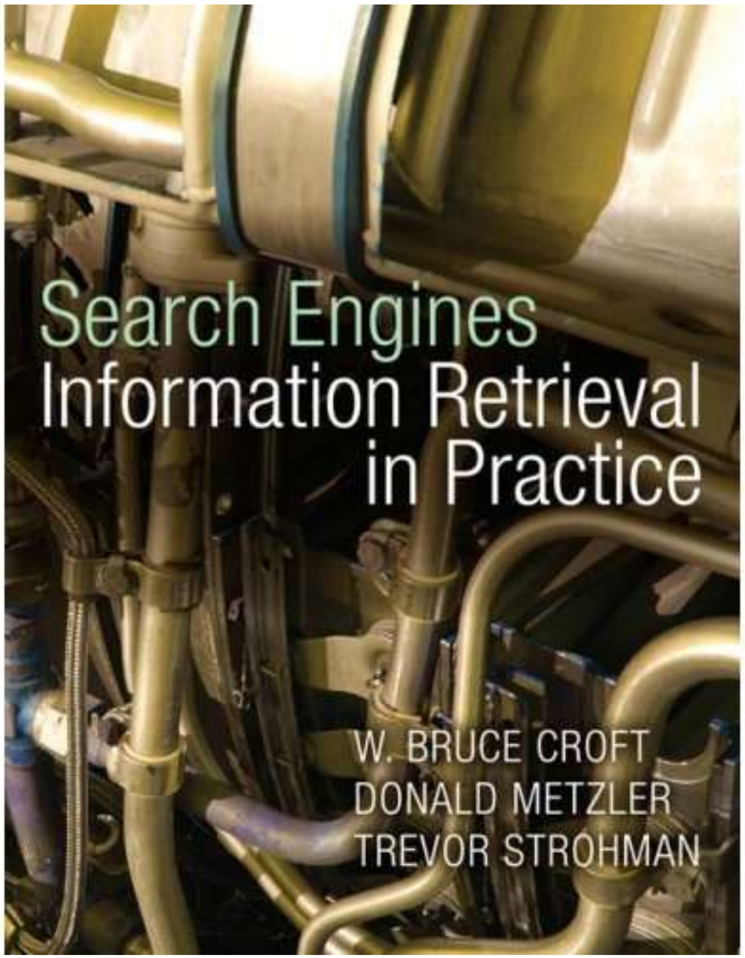
Example of Web Search Architecture



Component Technologies for Web IR

- Relevance Ranking
- Importance Ranking
- Document Understanding
- Query Understanding
- User Understanding
- Crawling
- Indexing
- Search Result Presentation
- Anti-Spam
- Search Log Data Mining

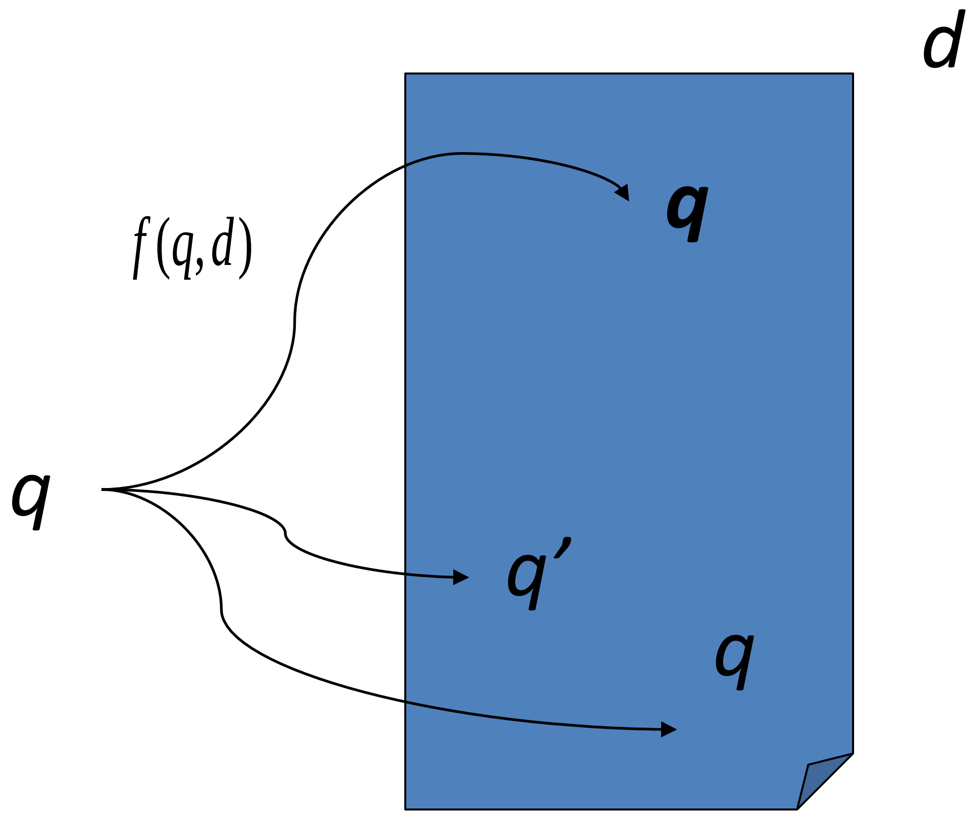
Book by Croft et al.



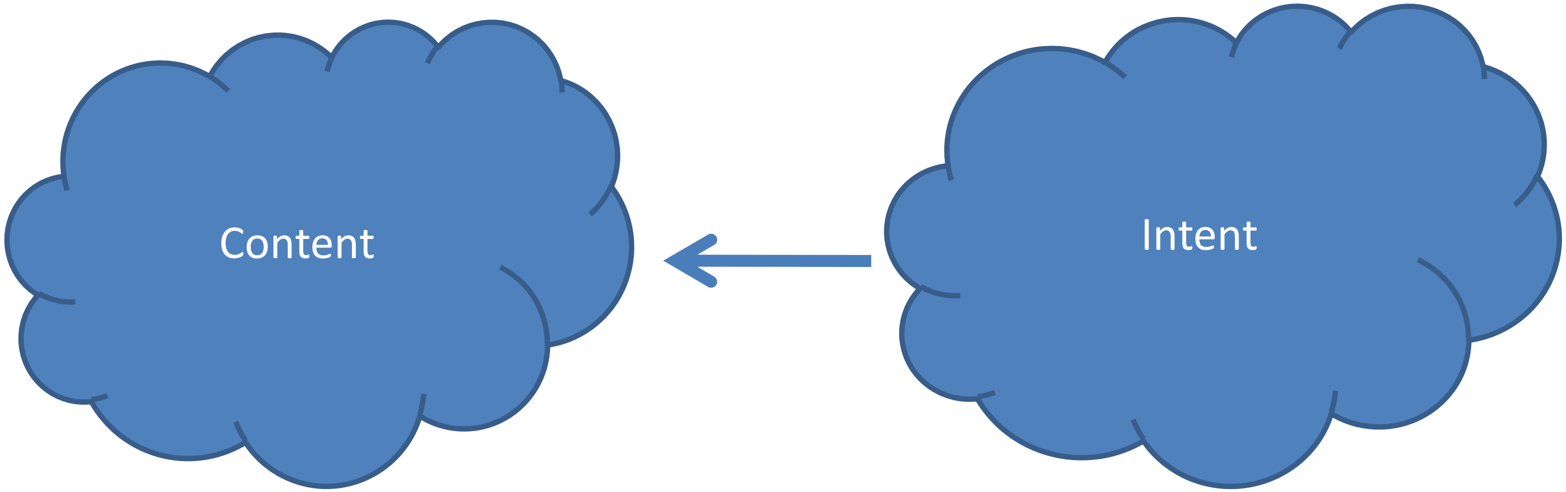
http://www.search-engines-book.com/

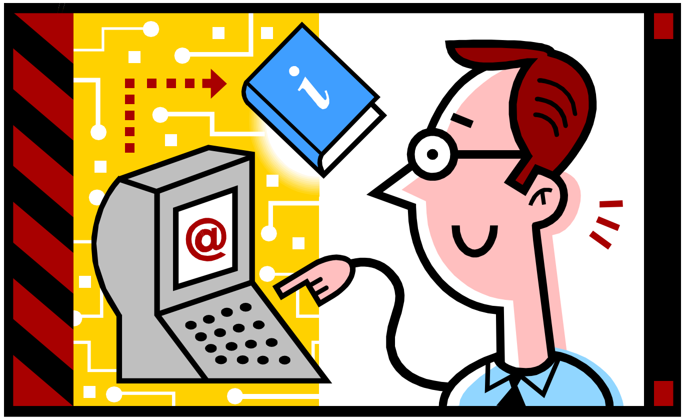
Relevance Model
(Matching Model)

Matching between Query and Document

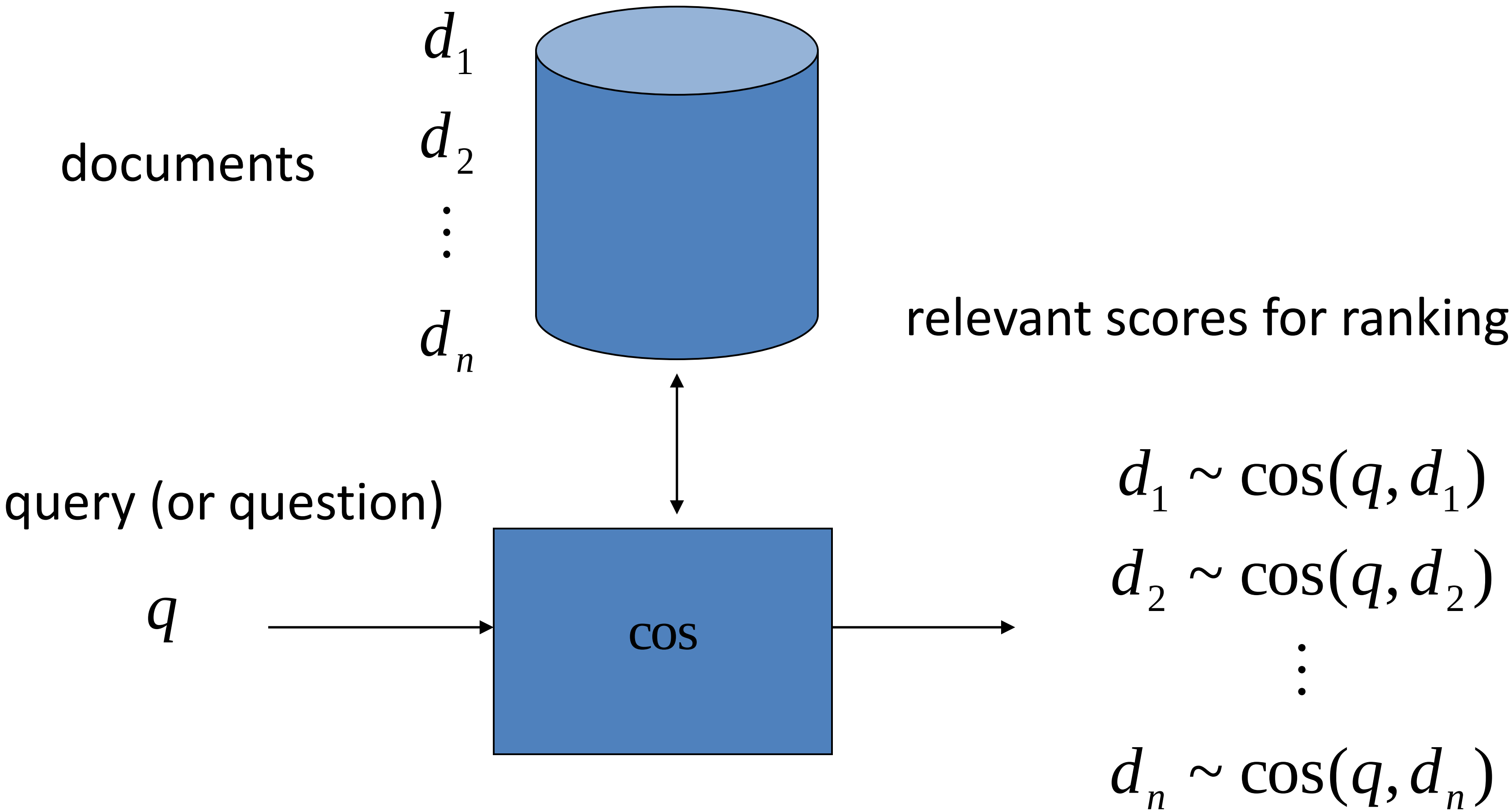


Matching between Two Worlds

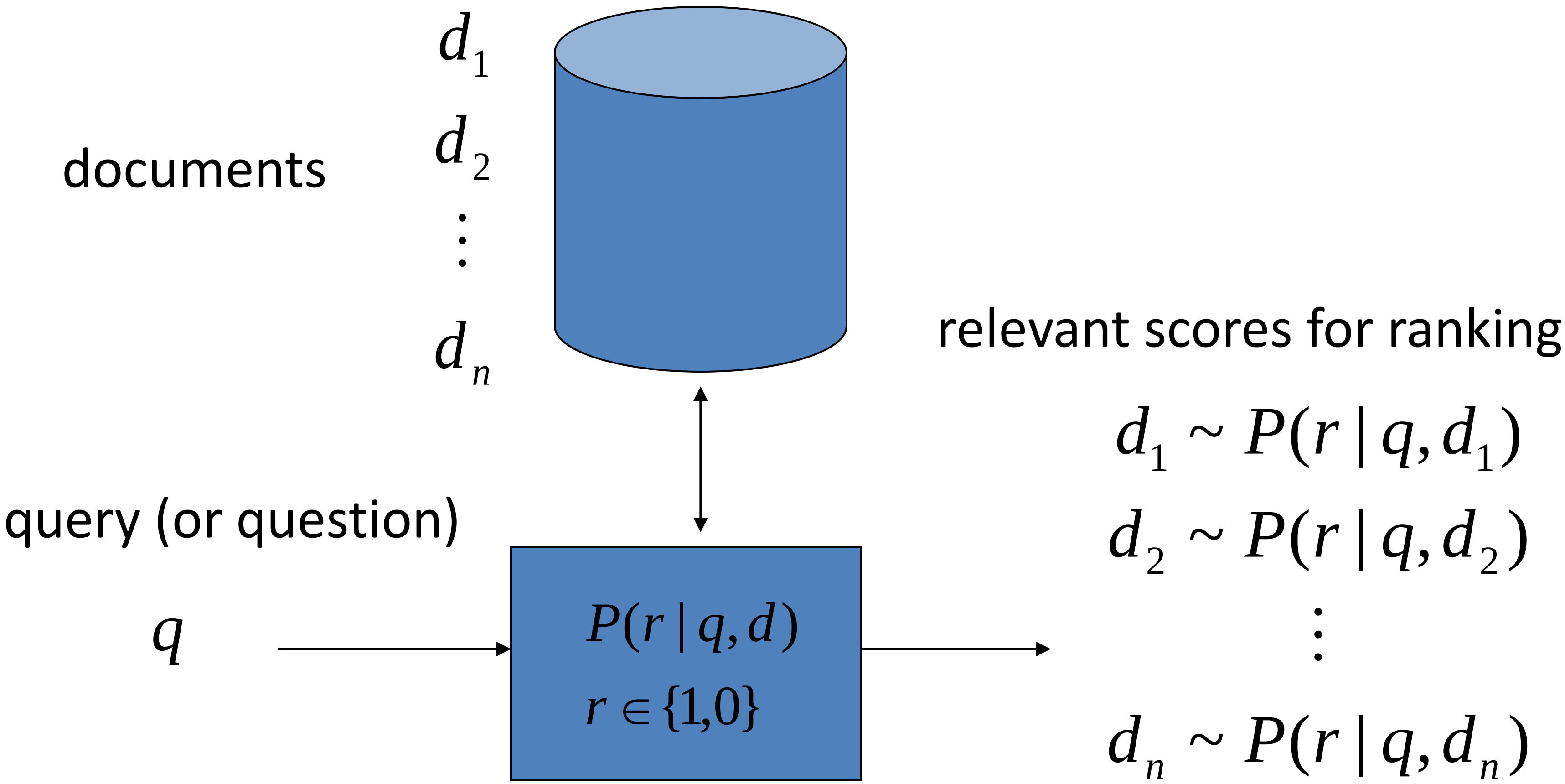




Vector Space Model (Salton 1975)

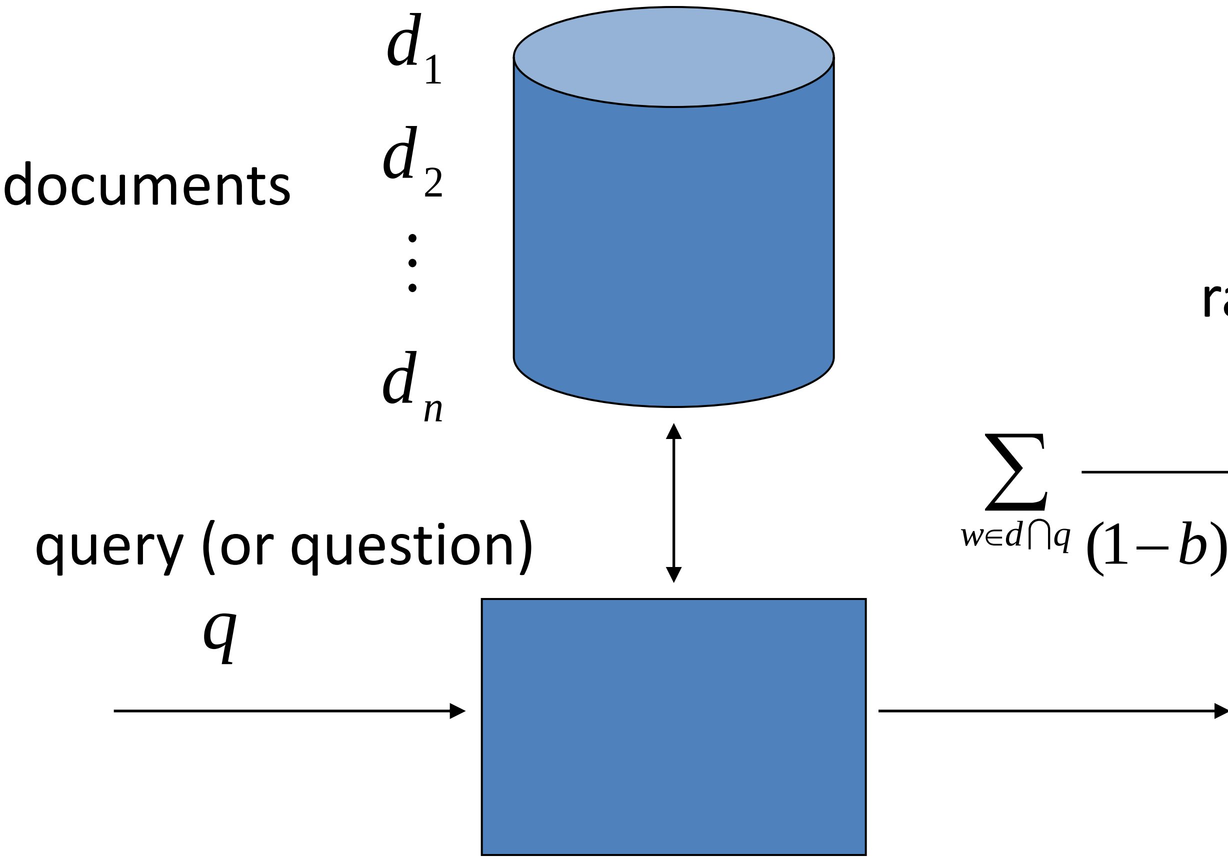


Probabilistic Model



Okapi or BM25

(Robertson and Walker 1994)



ranking function

$$\sum_{w \in d \cap q} \frac{(k+1)tf(w)}{(1-b)k + b\dfrac{dl}{avgdl} + tf(w)}$$

Language Mode

(Ponte and Croft 1998)

document = bag of words

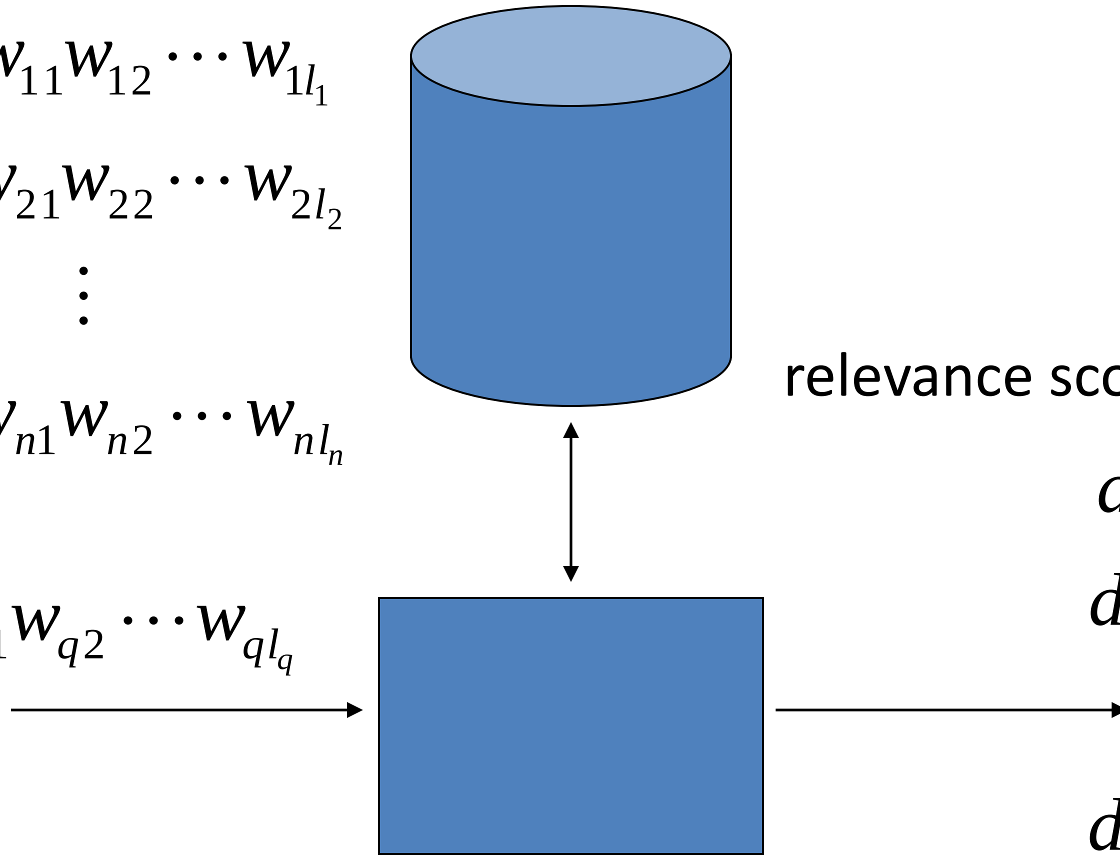
$$d_1 = w_{11} w_{12} \cdots w_{1l_1}$$

$$d_2 = w_{21} w_{22} \cdots w_{2l_2}$$

$$\vdots$$

$$d_n = w_{n1} w_{n2} \cdots w_{nl_n}$$

$$q = w_{q1} w_{q2} \cdots w_{ql_q}$$



relevance scores for ranking

$$d_1 \sim P(q \mid d_1)$$

$$d_2 \sim P(q \mid d_2)$$

$$\vdots$$

$$d_n \sim P(q \mid d_n)$$

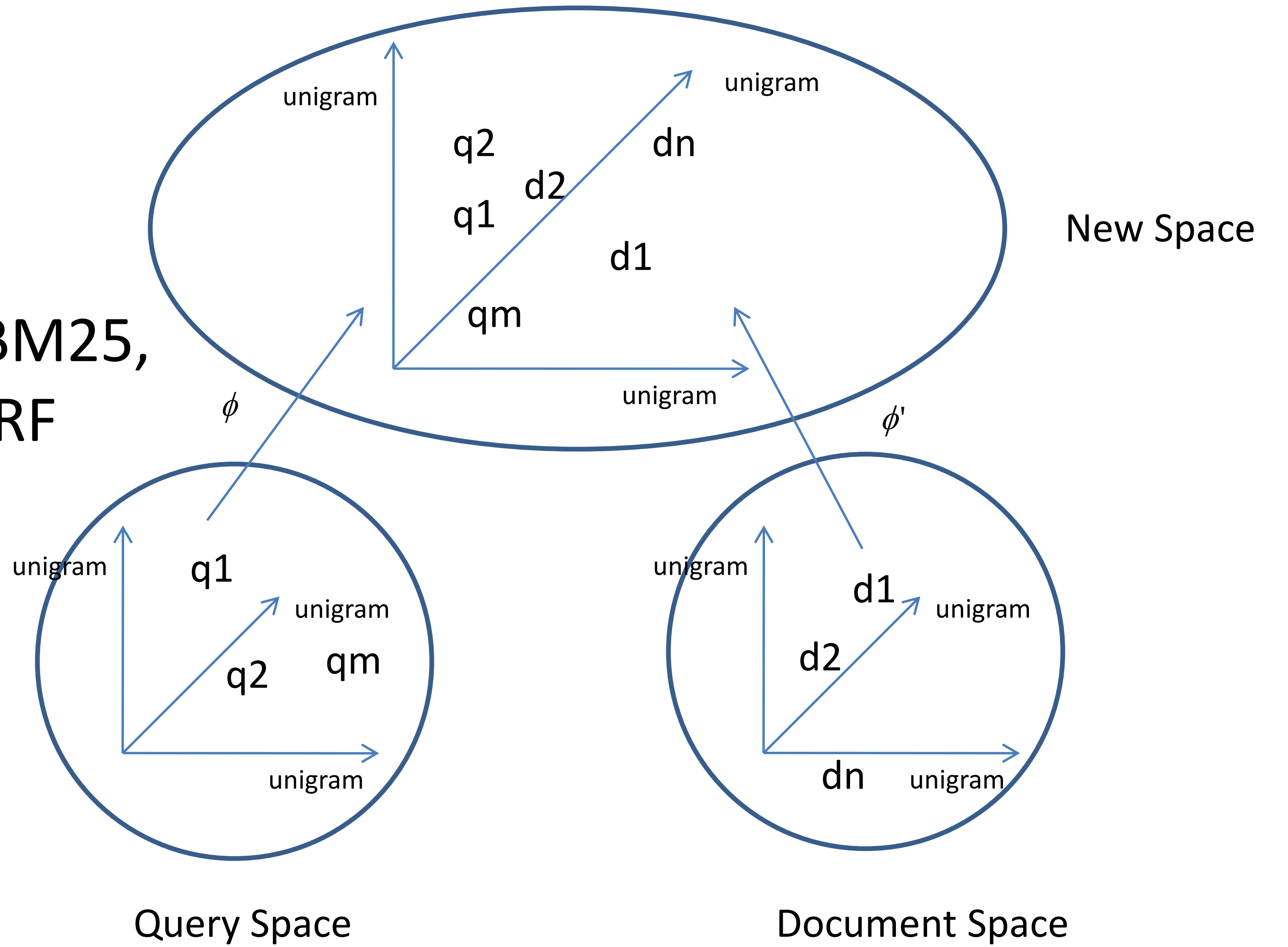
Relevance Model as Similarity Function

Jun Xu, Hang Li, Chaoling Zhong

AIRS 2010

IR Models as Similarity Functions (Similarity Functions)

VSM, BM25, LM, MRF



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 $w \in V$

    $\phi_Q^{BM25}(q)_w = \frac{(k_3+1)\times tf(w,q)}{k_3+tf(w,q)}$

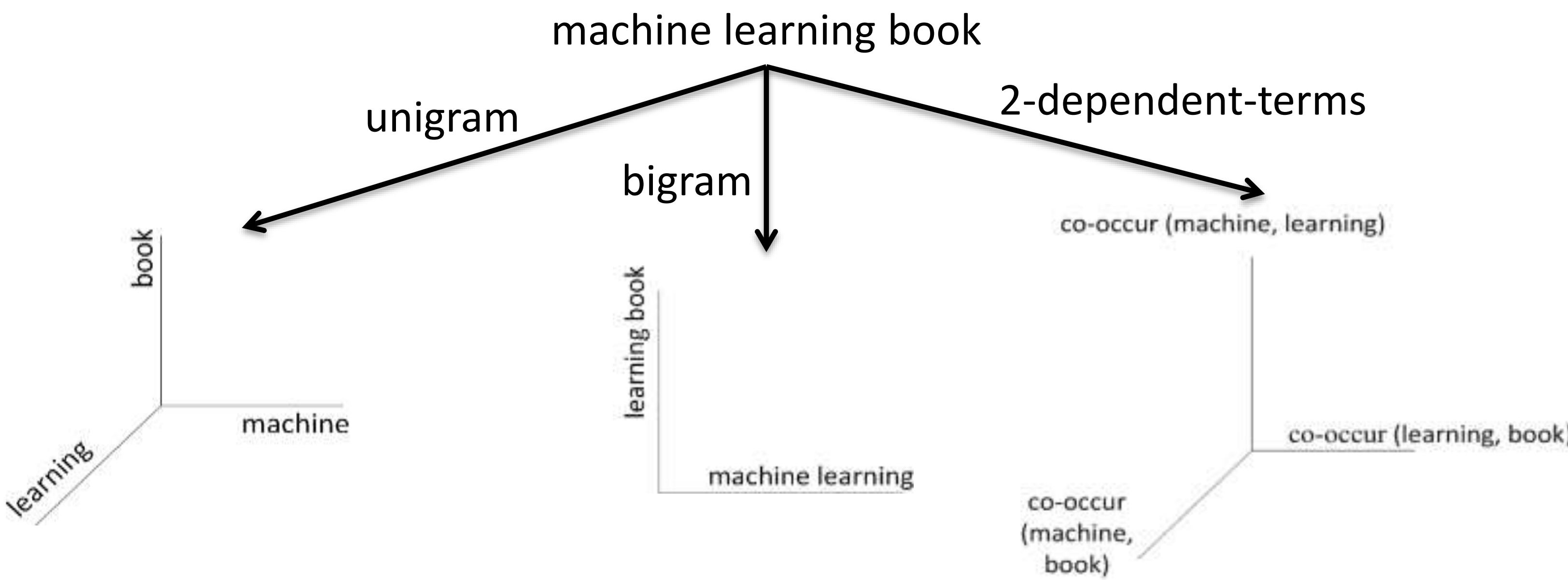
    $\phi_D^{BM25}(d)_w = \mathrm{IDF}(w) \cdot \frac{(k_1+1)\times tf(w,d)}{k_1\left(1-b+b\cdot\frac{len(d)}{avgDocLen}\right)+tf(w,d)}$

- 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

Relevance beyond Unigram



Extension of IR models

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

- BM25_Kernel
  - $\mathrm{BM25}\_Kernel(q,d) = \sum_t BM25\_Kernel_t(q,d)$ where t is dependence type
  - $\mathrm{BM25}\_Kernel_t(q,d) = \left\langle \phi_{Q,t}^{BM25}(q), \phi_{D,t}^{BM25}(d) \right\rangle$, and for all $x \in V_t$
    $\phi_{Q,t}^{BM25}(q)_x = \frac{(k_3+1)\times f_t(x,q)}{k_3+f_t(x,q)}$
    $\phi_{D,t}^{BM25}(d)_x = \mathrm{IDF}_t(x) \cdot \frac{(k_1+1)\times f_t(x,d)}{k_1\left(1-b+b\cdot\frac{f_t(d)}{avgDocLen_t}\right)+f_t(x,d)}$

Similarity Function

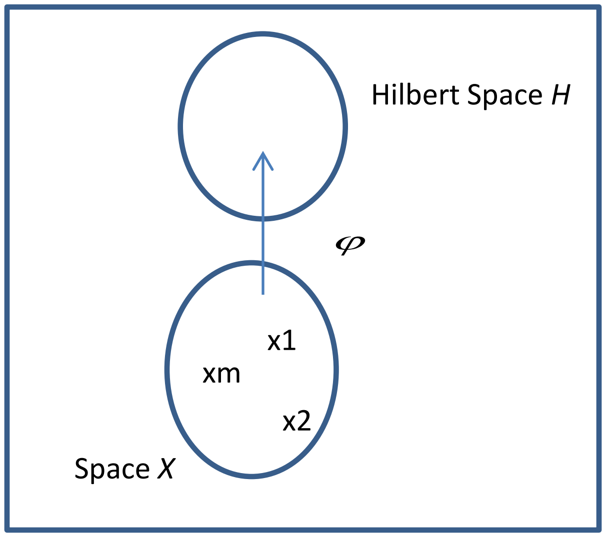
- Kernel $k: \mathcal{X} \times \mathcal{X} \to \Re$
  - Definition: $k(x, x') = \langle \phi(x), \phi(x') \rangle, where\ \phi: \mathcal{X} \to \mathcal{H}$
  - Given k_1 and k_2 are kernels, create new kernels: $\alpha k, where\ \alpha \geq 0$; $k_1 + k_2$; $k_1 \cdot k_2$

- Similarity function: $k: \mathcal{X} \times \mathcal{Y} \to \Re$
  - Definition: $k(x, y) = \langle \phi(x), \phi'(y) \rangle, where\ \phi: \mathcal{X} \to \mathcal{H}$ and $\phi': \mathcal{Y} \to \mathcal{H}$
  - Given k_1 and k_2 are similarity functions, create new similarity functions:

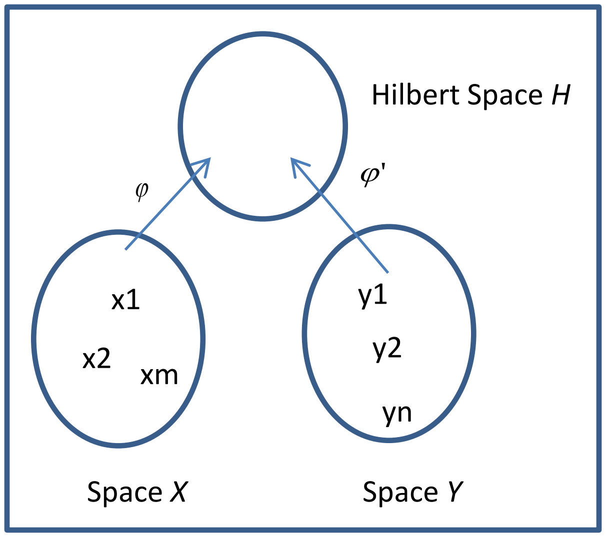
    $\alpha k, where\ \alpha \in \Re$; $k_1 + k_2$; $k_1 \cdot k_2$

Kernel vs Similarity Function

$$k(x, x') = \langle \phi(x), \phi(x') \rangle$$



$$k(x, y) = \langle \phi(x), \phi'(y) \rangle$$



Query Document Mismatch

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

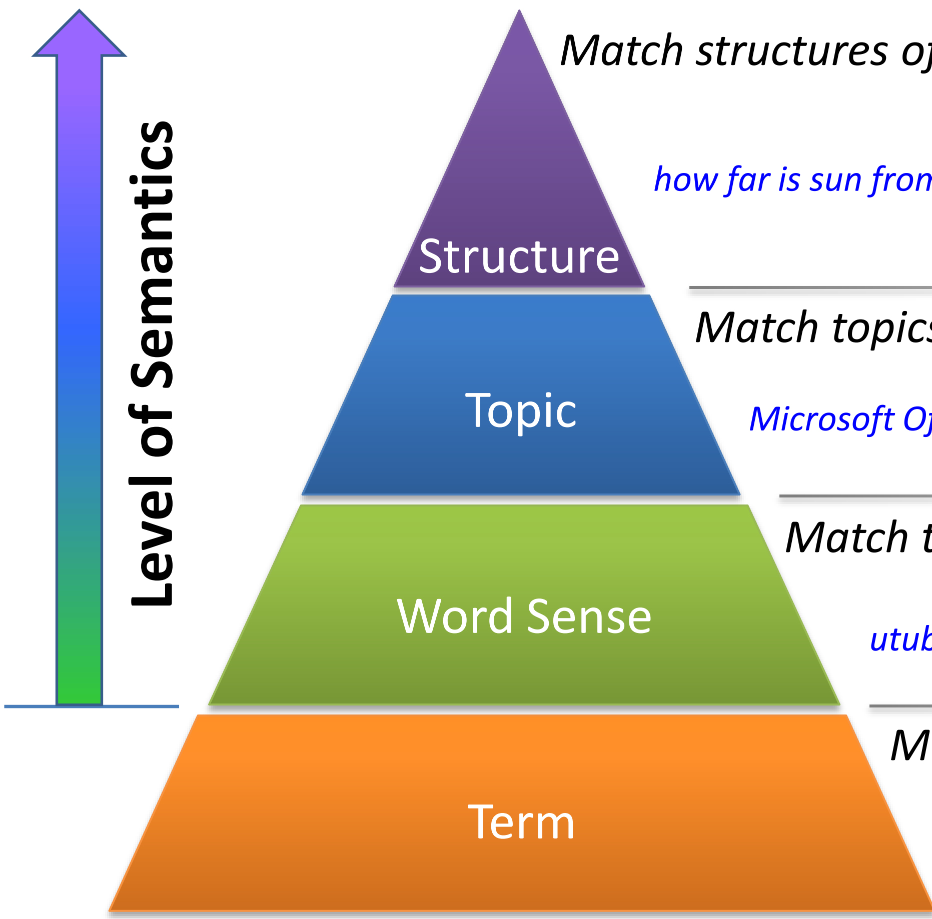
Same Search Intent, Different Query Representations Example = “Youtube”

• 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

Examples of Term Mismatch

- **Query → Document**
- swimming pool schedule = pool schedule
- seattle best hotel = seattle best hotels
- natural logarithm transformation = logarithm transformation
- china kong ≠ china hong kong
- why are windows so expensive ≠ why are macs so expensive

Different Levels of Semantic Matching



Match structures of query and document

how far is sun from earth → *… distance between sun and earth*

Match topics of query and document

Microsoft Office → *… Microsoft … PowerPoint, Word, Excel…*

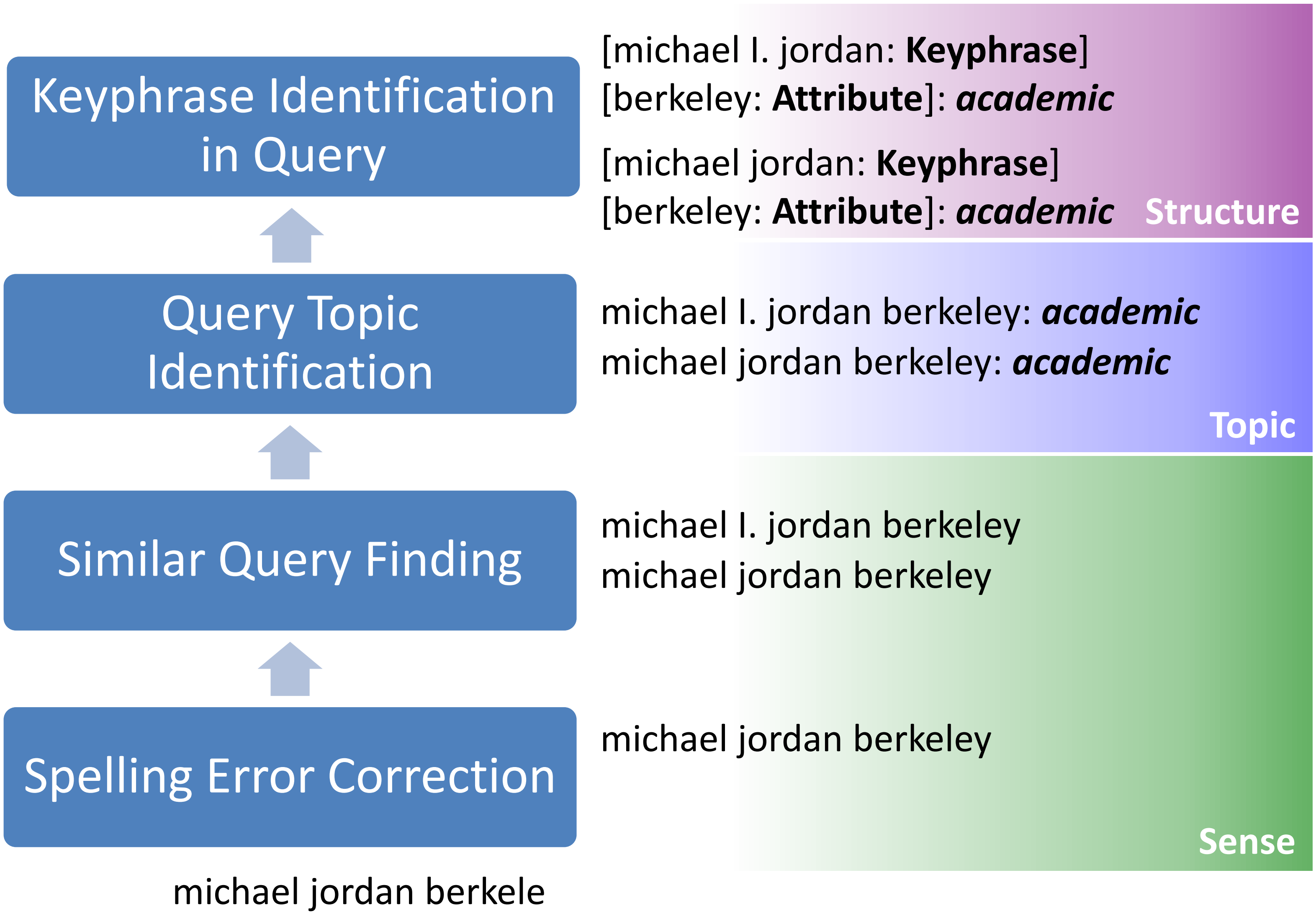
Match terms with same meanings

utube → *youtube* *NY* → *New York*

Match exactly same terms

NY → *NY* *youtube* → *youtube*

Query Understanding (Online)



Document Understanding (Offline)

Document Topic Identification

↑

Keyphrase Identification

↑

Tokenization

Michael Jordan is Professor in the Department of Electrical Engineering

[***Michael Jordan/M. Jordan***] *is* [***Professor***] *in the* [*Department/Dept.*] *of* [***Electrical Engineering/EE***]: ***academic***

Topic

[***Michael Jordan/M. Jordan: Keyphrase***] *is* [***Professor***] *in the* [*Department/Dept.*] *of* [***Electrical Engineering/EE***]

Structure

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

Term

Online Semantic Matching


Michael I. Jordan's Home Page
Models of visuomotor and other learning (Univ. of California, Berkeley, USA)
www.cs.berkeley.edu/~jordan · Cached page · Mark as spam

Michael Jordan | EECS at UC Berkeley
Michael Jordan Professor Research Areas Artificial Intelligence (AI) Biosystems & Computational Biology (BIO) Control, Intelligent Systems, and Robotics (CIR)
www.eecs.berkeley.edu/Faculty/Homepages/jordan.html · Cached page · Mark as spam

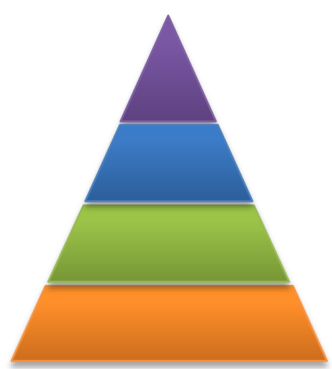
Publications
Jordan. In M.-H. Chen, D. Dey, P. Mueller, D. Sun, and K. Ye (Eds.), Frontiers of ...
Technical Report 661, Department of Statistics, University of California, Berkeley, 2004.
www.cs.berkeley.edu/~jordan/publications.html · Cached page · Mark as spam

Ranking Results

Query Representation → **Semantic Matching** ← **Document Representation**

[michael I. jordan: **Keyphrase**]
[berkeley: **Attribute**]: ***academic***

[michael jordan: **Keyphrase**]
[berkeley: **Attribute**]: ***academic***



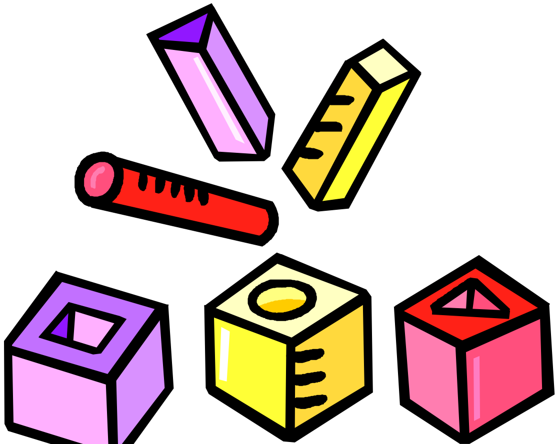
*[**Michael Jordan/M. Jordan: Keyphrase**] is [**Professor**] in the [Department/Dept.] of [**Electrical Engineering/EE**]:* ***academic***

Matching can be conducted at different levels

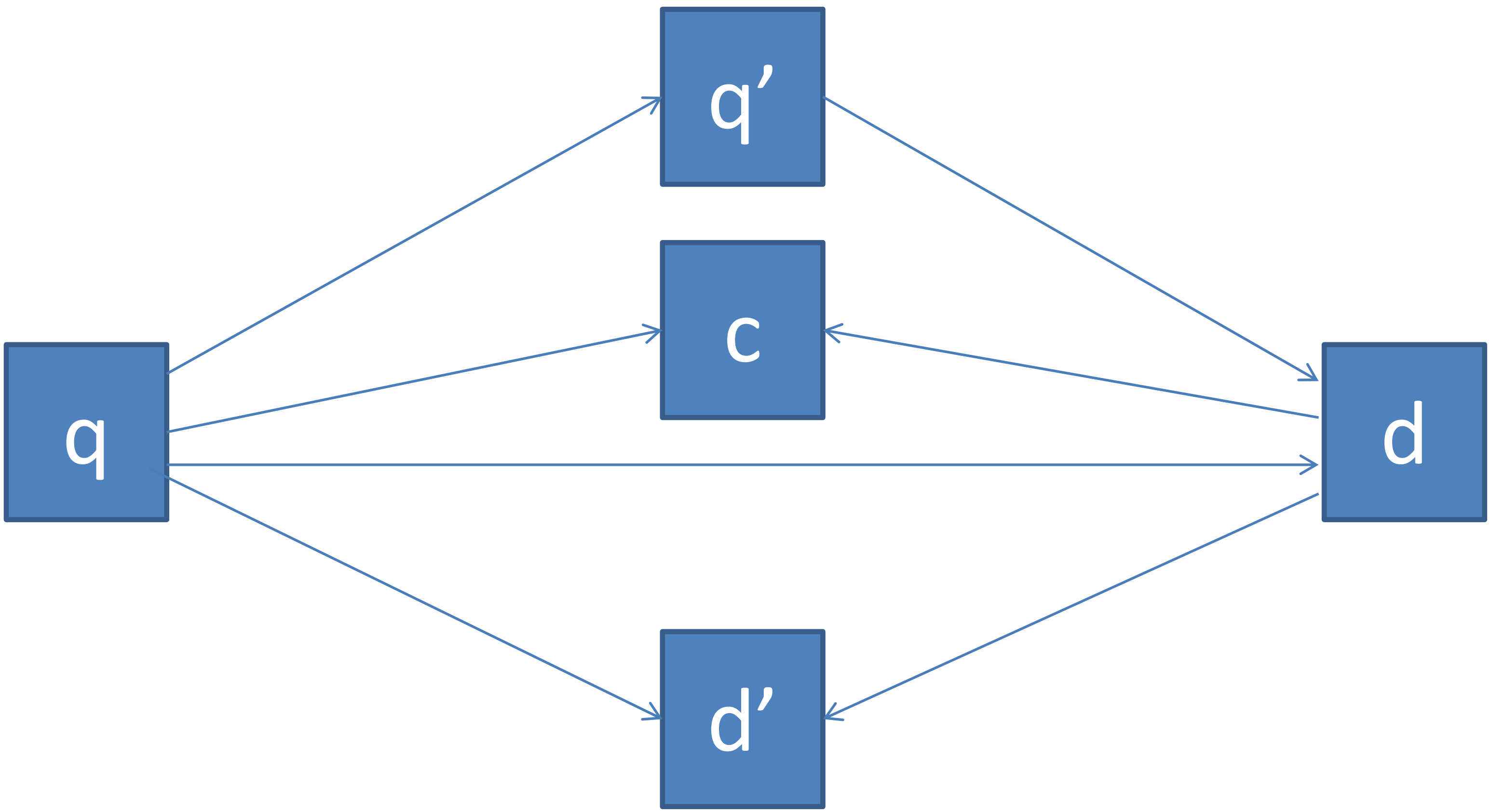
Related Work

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

Learning to Match



Four Ways to Match



Learning to Match

- Learning matching function

$$f_M(q, d)$$

- Using training data $(q_1, d_1), \cdots, (q_N, d_N)$
- *Using prior knowledge or other data M*

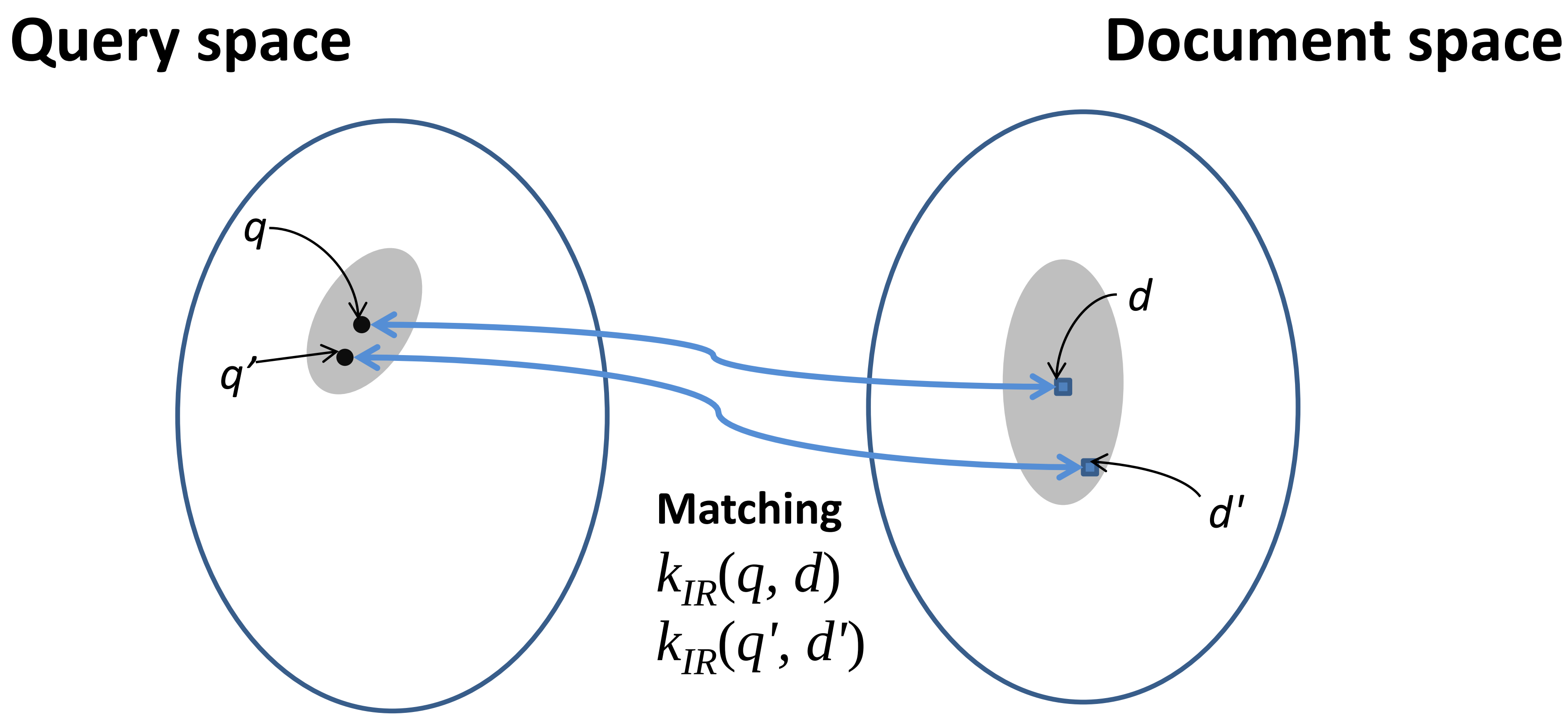
Challenges in Matching

- How to incorporate prior knowledge or other data into model
- Scale is very large

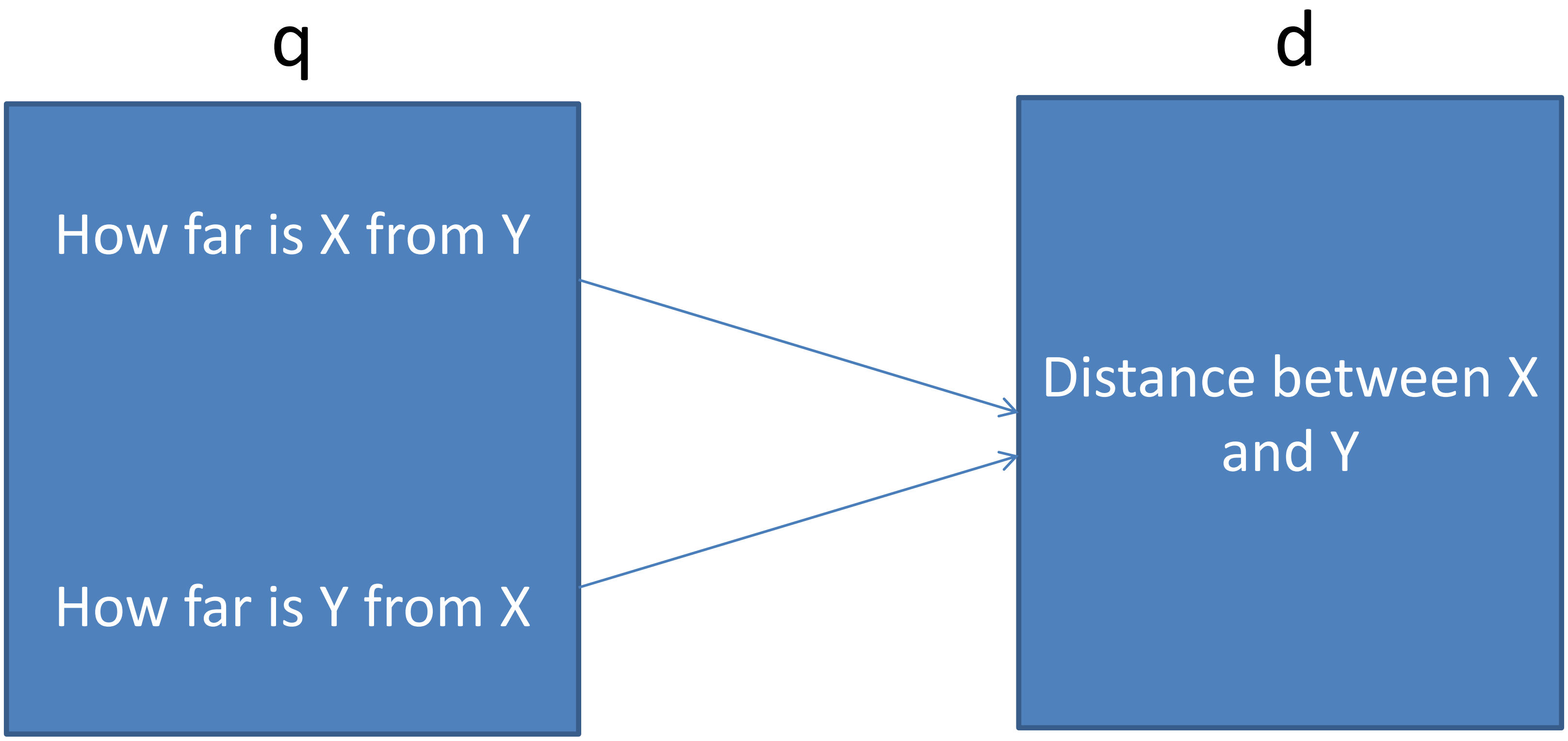
Matching Problem: Matrix Data View

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

Matching Problem: Space View



Matching Problem: String Data View



Examples of Matching Models

- Similarity Learning

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

- Topic Modeling

$$\boxed{f_M(q,d) = \sum_k u(q,k) v(k,d)}$$

- String Transformation

$$f_M(q,d) = f(q,d) + \sum_i \boxed{k_T(q,q_i)} k_T(d,d_i) f(q_i,d_i)$$

Matching vs Ranking

	Matching	Ranking
Prediction	Matching score between query and document	List of documents
Model	f(q, d)	f(q,d1), f(q,d2), … f(q,dn)
Loss Function	Single query document pair	List of documents with respect to query
Challenge	Mismatch	Correct ranking on top

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)

Our Methods of Learning to Match

Three Approaches of Learning to Match

- Similarity Learning → Word sense level
- Topic Modeling → Topic level
- String Transformation → Structure level

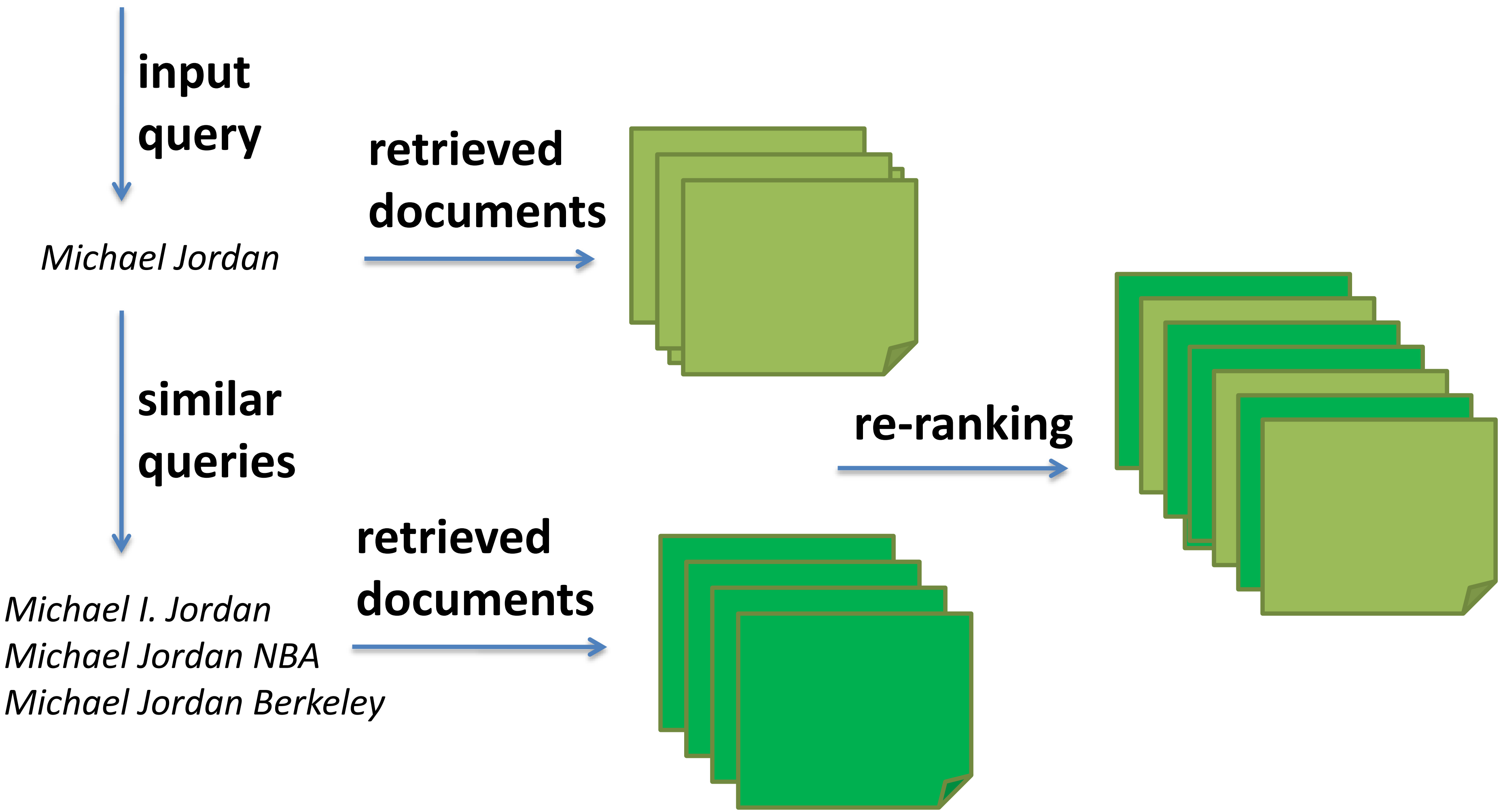
Summary of Our Technologies

	Technologies	Current Status
Term matching	Key n-gram learning	ongoing
Term matching	Relevance model as similarity function	AIRS'10 best paper
Sense matching	Robust relevance model	JMLR, WWW'11 poster
Sense matching	Query similarity learning	WSDM'11
Sense matching	CRF model for candidate selection	SIGIR'08
Sense matching	Log linear model for candidate generation	ACL'11
Sense matching	Projection to latent structure	ongoing
Topic matching	Scalable and efficient topic modeling	SIGIR'11

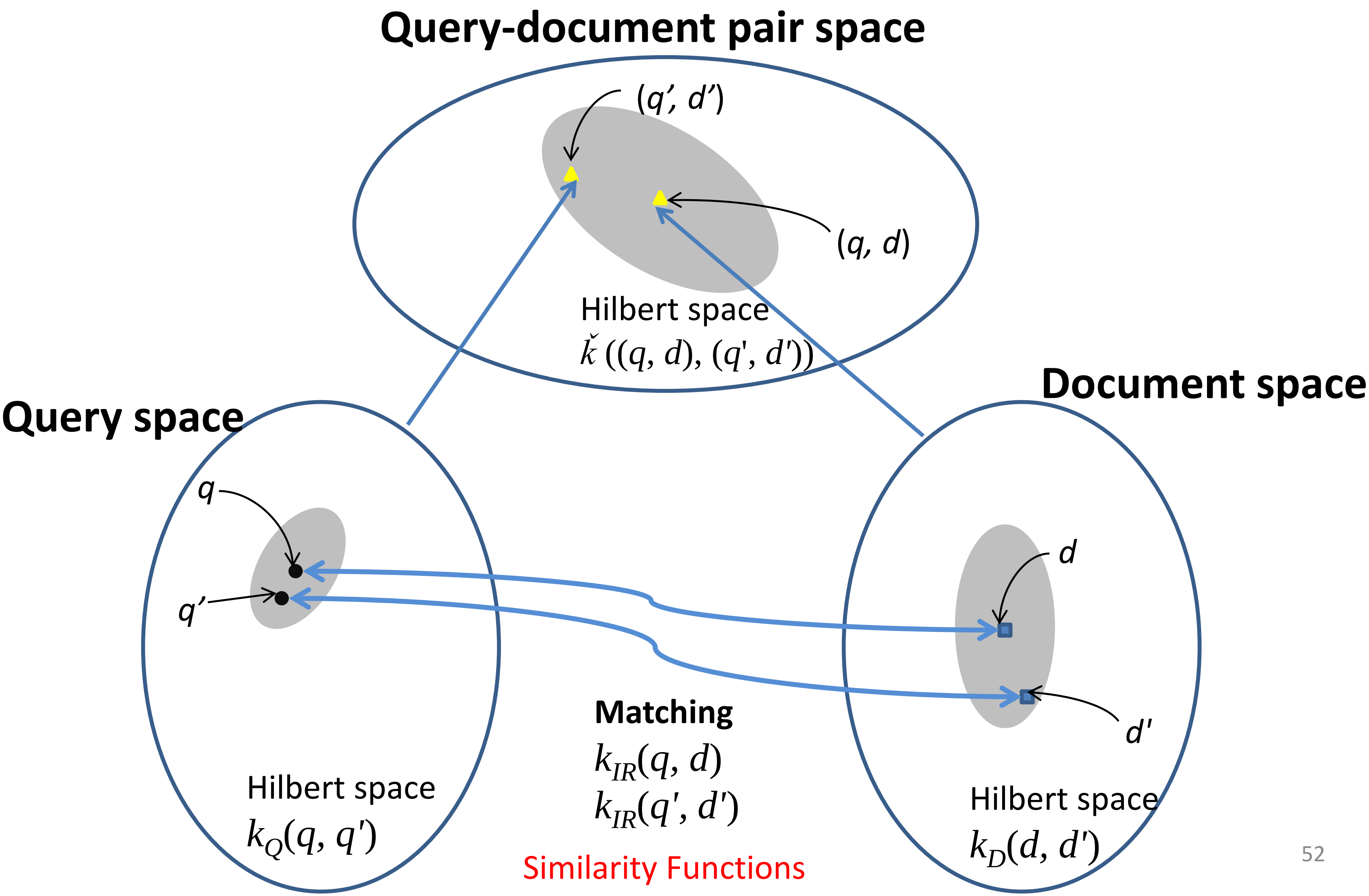
Robust Similarity Function Learning Using Kernel Methods

Wei Wu, Hang Li, Jun Xu, Satoshi Oyama, JMLR 2011

Dealing with Mismatch with Re-Ranking
- Our Approach = Online Learning of Kernel Methods



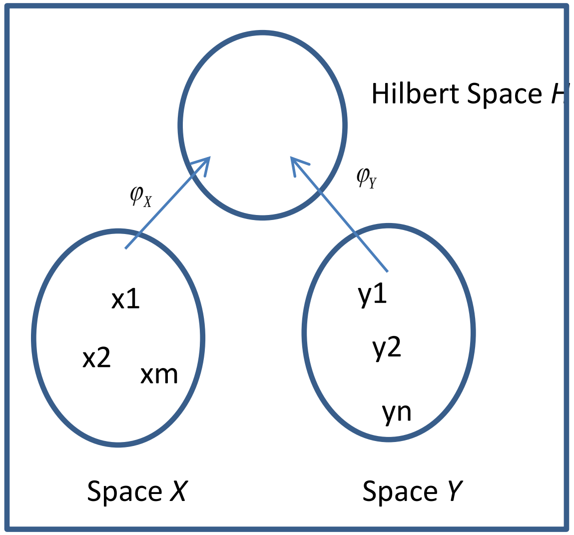
Mapping to Space of Query Document Pairs - Using Kernel Methods



Similarity Learning

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

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



Similarity Learning Using Kernel Methods

- Assumption
  - Space of similarity functions is RKHS generated by positive-definite kernel $\bar{k}$: $(\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y})$
  -
- Optimization

$$\min_{k \epsilon \mathcal{K}} \frac{1}{N} \sum_{i=1}^{N} l(k(x_i, y_i), t_i) + \frac{\lambda}{2} \|k\|_{\mathcal{K}}{}^2$$

- Solution
  - By representer theorem $k^*(x, y) = \sum_{i=1}^{N} \alpha_i \bar{k}((x_i, y_i), (x, y))$
-

$$\bar{k}\big((x, y), (x', y')\big) = g(x, y) k_{\mathcal{X}}(x, x') k_{\mathcal{Y}}(y, y') g(x', y')$$

Learning Robust BM25

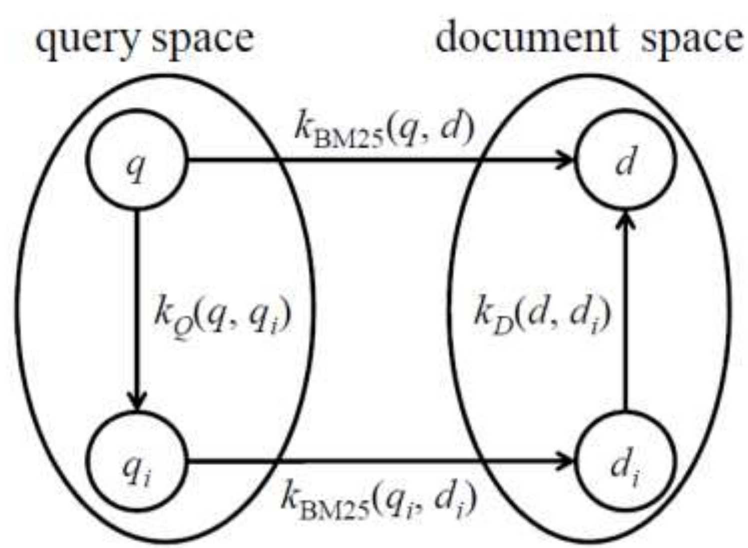
- BM25 :
- Kernel

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

- Solution (called Robust BM25)

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

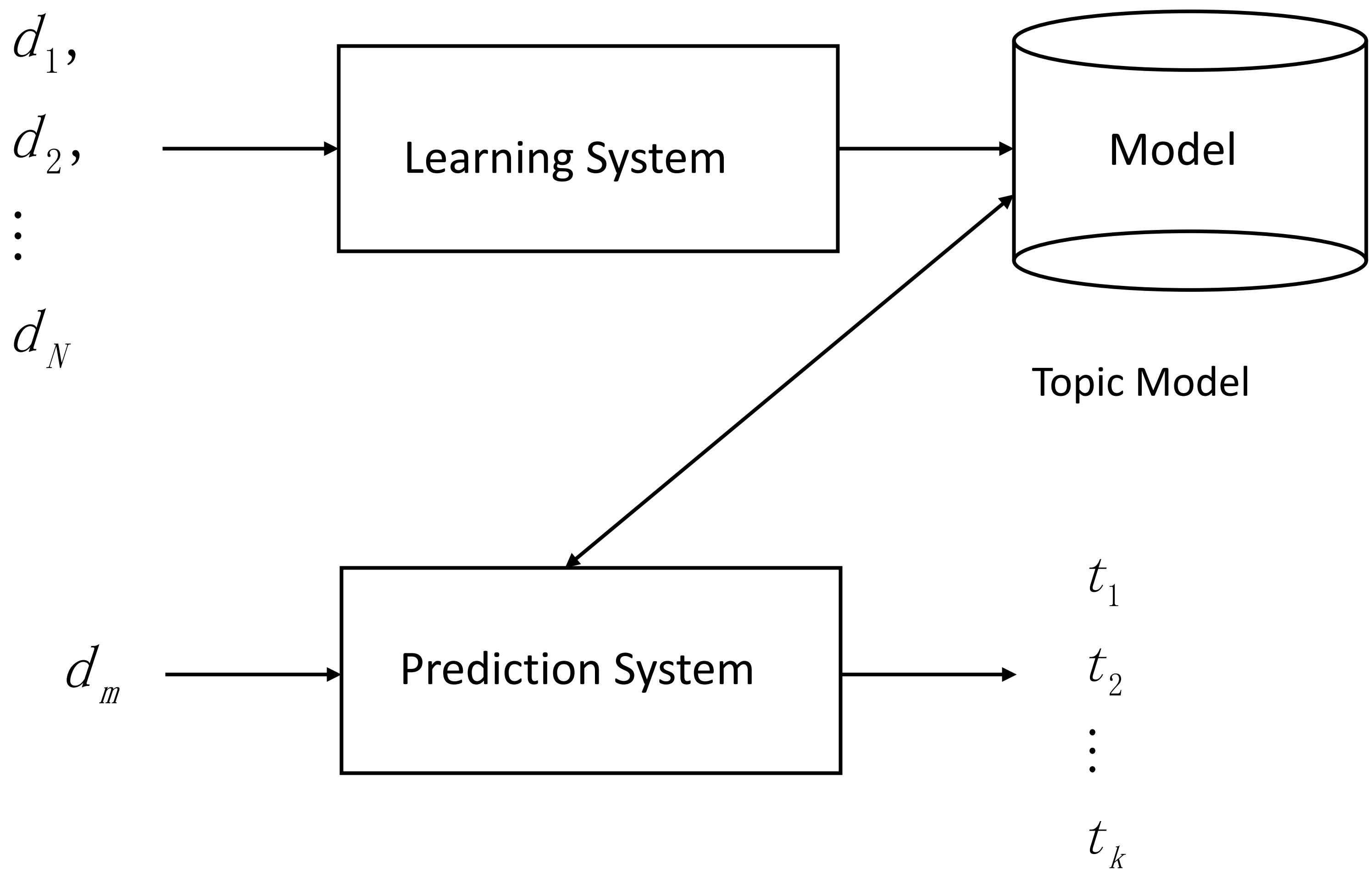
- Deal with term mismatch



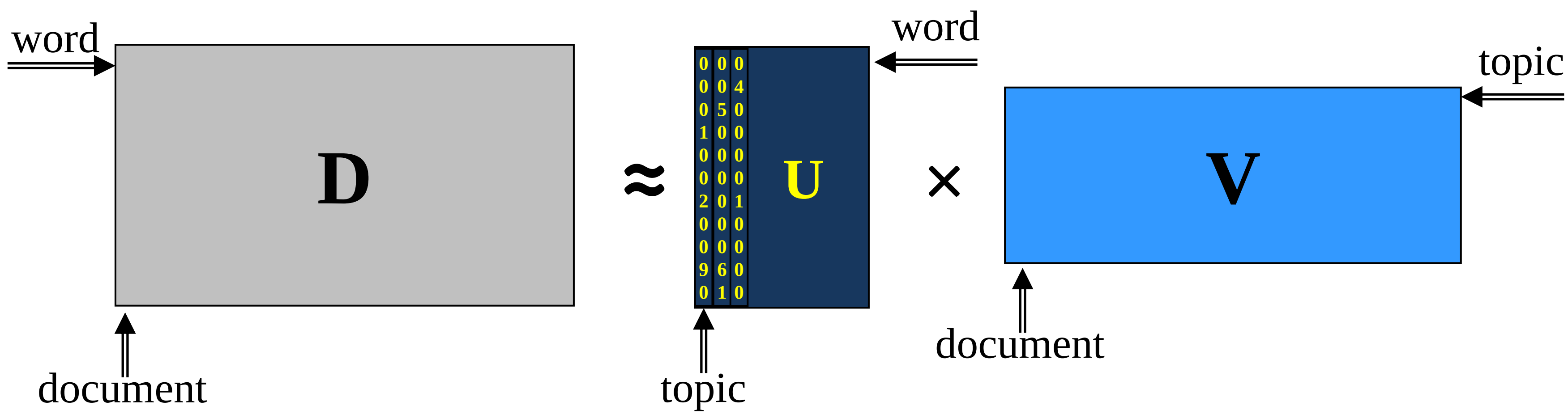
Regularized Latent Semantic Indexing

Quan Wang, Jun Xu, Hang Li, Nick Craswell, SIGIR 2011

Topic Modeling
Our Approach = Regularized Latent Semantic Indexing



Regularized Latent Semantic Indexing



word representation of doc n

topic matrix

topic representation of doc n

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

topics are sparse

Regularized Latent Semantic Indexing

- L1 on topics and L2 on documents
- L1 leads to sparse topics, a topic only contains a small number of words
- L2 leads to accurate modeling
- Formulation is simple
- Easy to scale up

Scalability Comparison

algorithm	max dataset applied (#docs; #words)	# topics	# processors used
PLDA and PLDA+ (by Google)	Wiki-200T (2,112,618; 200,000)	1000	2,048
AD-LDA (by UCI)	NY Times (300,000; 102,660)	200	16
RLSI	B01 (1,562,807; 7,014,881) Wikipedia (3,239,884; 1,689,193) Bing News (1,028,070; 940,702)	500 ~ 1000	16 single machine!

Regularized Topics

AP dataset, topic compactness: 0.0075

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

drug	soviet	aid	court	Jackson	percent	student	nuclear	Bush	Israel
cocaine	Afghanistan	virus	senate	Dukaki	billion	Korea	soviet	Dukaki	Palestinian
traffick	Afghan	infect	Reagan	democrat	rate	protest	treaty	campaign	Israeli
test	Gorbachev	test	house	delegate	0	Korean	missile	Quayle	Arab
enforce	Pakistan	patient	state	percent	trade	Chun	weapon	Bentsen	PLO

Optimization

Algorithm 2 Update $\mathbf{U}$

Require: $\mathbf{D} \in \mathbb{R}^{M \times N}$, $\mathbf{V} \in \mathbb{R}^{K \times N}$

```
S ← VV^T
R ← DV^T
for m = 1 : M do
    ū_m ← 0
    repeat
        for k = 1 : K do
            w_mk ← r_mk − Σ_{l≠k} s_kl u_ml
            u_mk ← (|w_mk| − ½ λN)_+ sign(w_mk) / s_kk
        end for
    until convergence
end for
return U
```

words processed in parallel

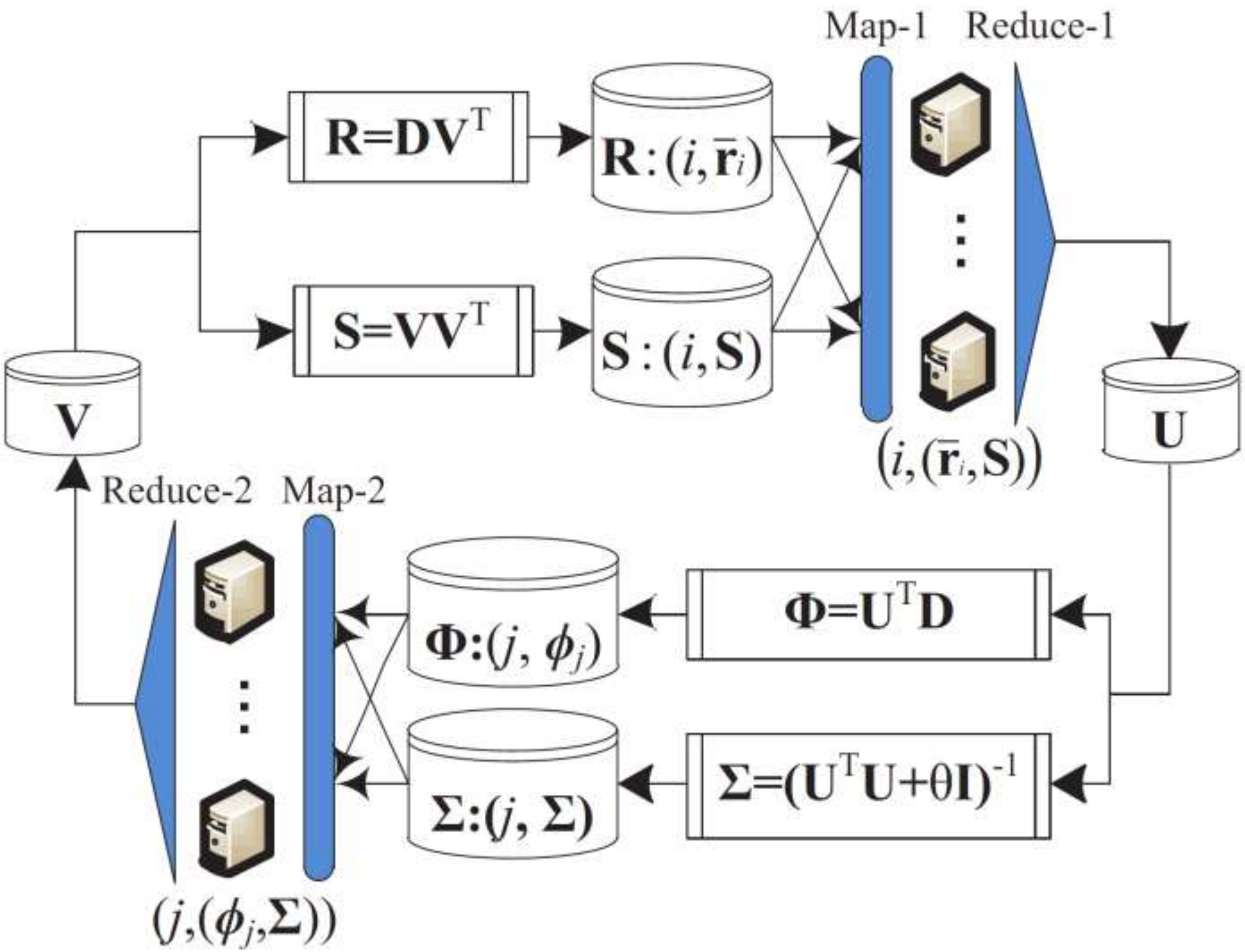
Algorithm 3 Update $\mathbf{V}$

Require: $\mathbf{D} \in \mathbb{R}^{M \times N}$, $\mathbf{U} \in \mathbb{R}^{M \times K}$

```
Σ ← (U^T U + θI)^{-1}
Φ ← U^T D
for n = 1 : N do
    v_n ← Σφ_n, where φ_n is the n^th column
end for
return V
```

docs processed in parallel

Scaling up on MapReduce

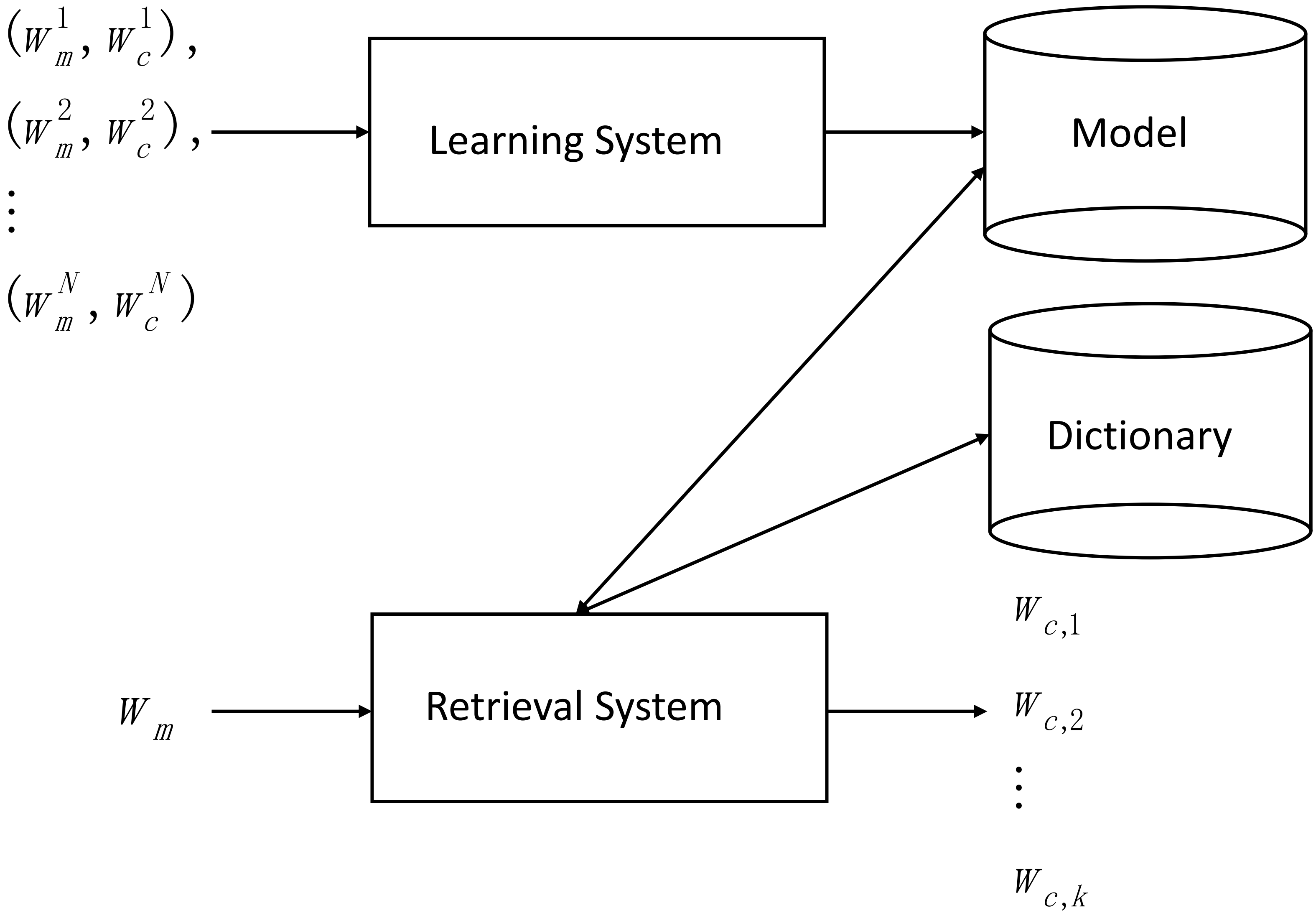


Query Generation Using Log Linear Model

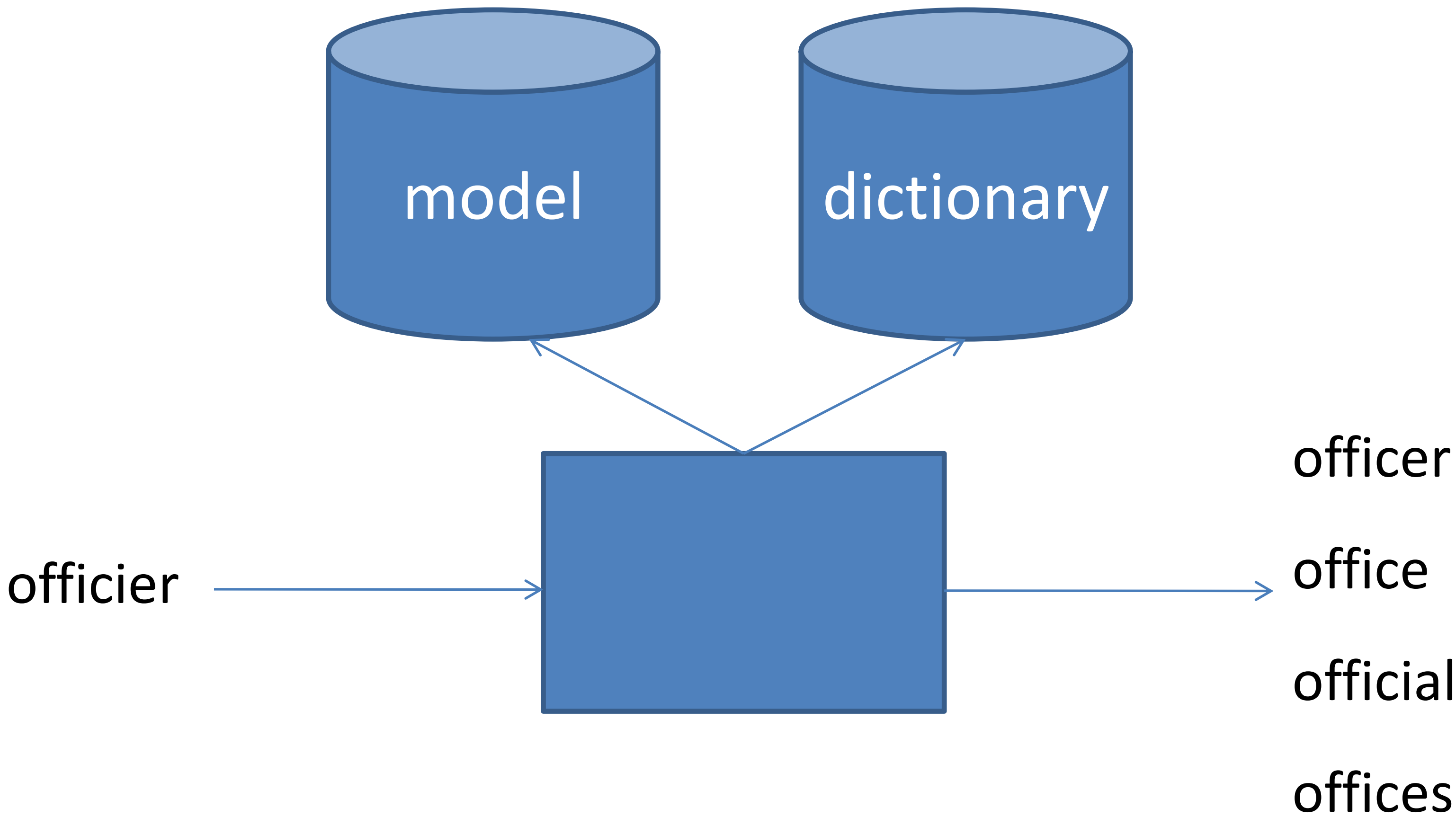
Ziqi Wang, Gu Xu, Hang Li, Ming Zhang

ACL 2011

Candidate Generation in Spelling Correction Our Approach = Log Linear Model



Candidate Generation in 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

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

Weight of rule

Set of rules rewrite w_m to w_c

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

$$\forall \lambda_r \leq 0$$

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

- Candidate Generation

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

Model Learning

- Objective function

$$\lambda^* = \arg\max_{\lambda} \sum_i \max_{R(w_m^i, w_c^i)} \log P(w_c^i, R(w_m^i, w_c^i) \mid w_m^i)$$

Take max over transformations

- Algorithm
  - Constrained Quasi Newton Method (BFGS)

Retrieval

Misspelled word → **Finding all matching rules** → **Find best candidates** → **Top k candidates**

W_m

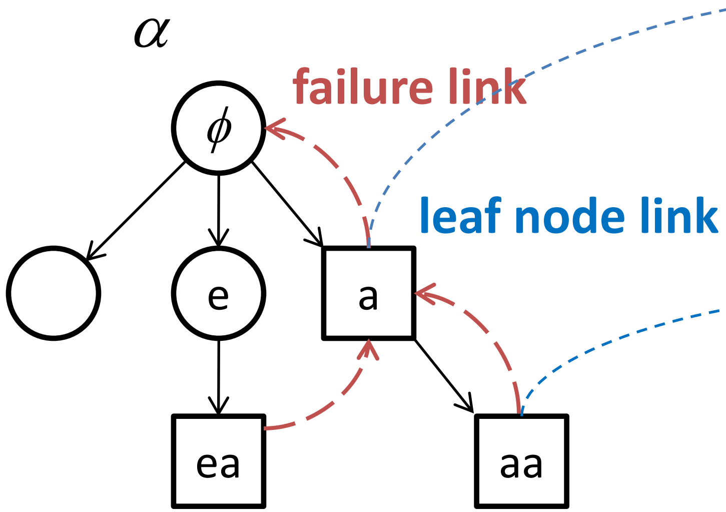
$$\alpha_1 \rightarrow \beta_1, \lambda_1$$
$$\alpha_2 \rightarrow \beta_2, \lambda_2$$
$$\alpha_3 \rightarrow \beta_3, \lambda_3$$
$$\ldots$$

$$w_{c,1}$$
$$w_{c,2}$$
$$w_{c,3}$$
$$\ldots$$

Aho Corasick Tree
(rule set)

Trie Tree
(dictionary)

Aho Corasick Tree



β

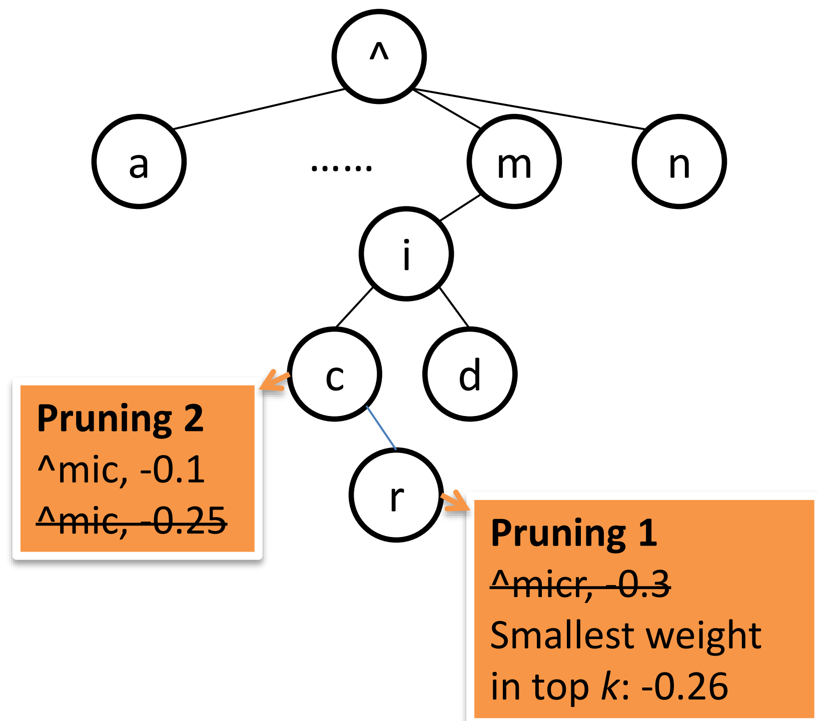
e	0.0	$a \rightarrow e$
s	-0.3	$a \rightarrow s$
……	……	
NULL	—	
a	-0.1	$aa \rightarrow a$
……	……	
NULL	—	
……	……	

Index all the α's in the rules on the AC tree

β are stored in an associated list

Retrieval with Dynamic Programming

- 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

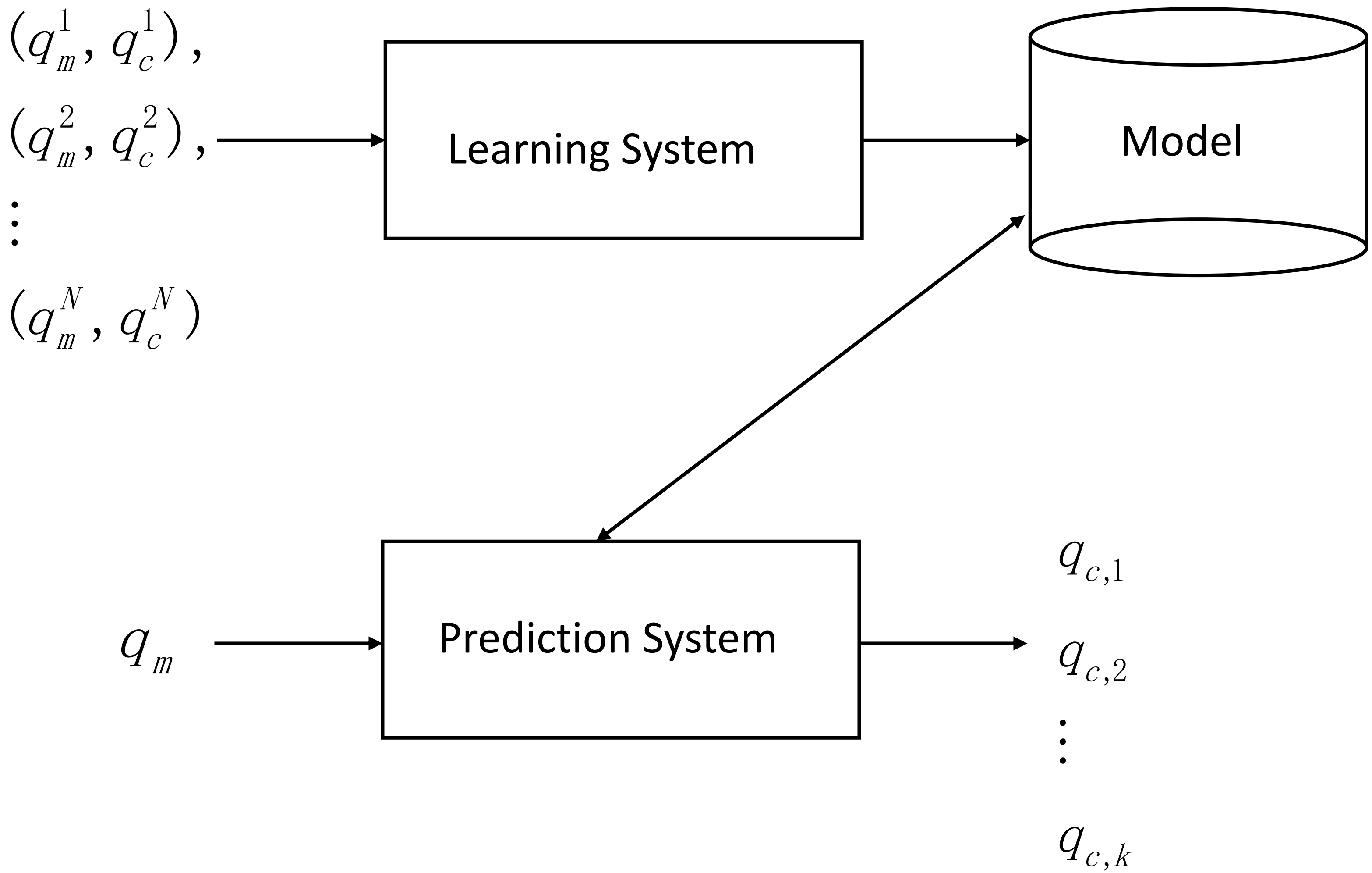


Query Rewriting Using Conditional Random Fields

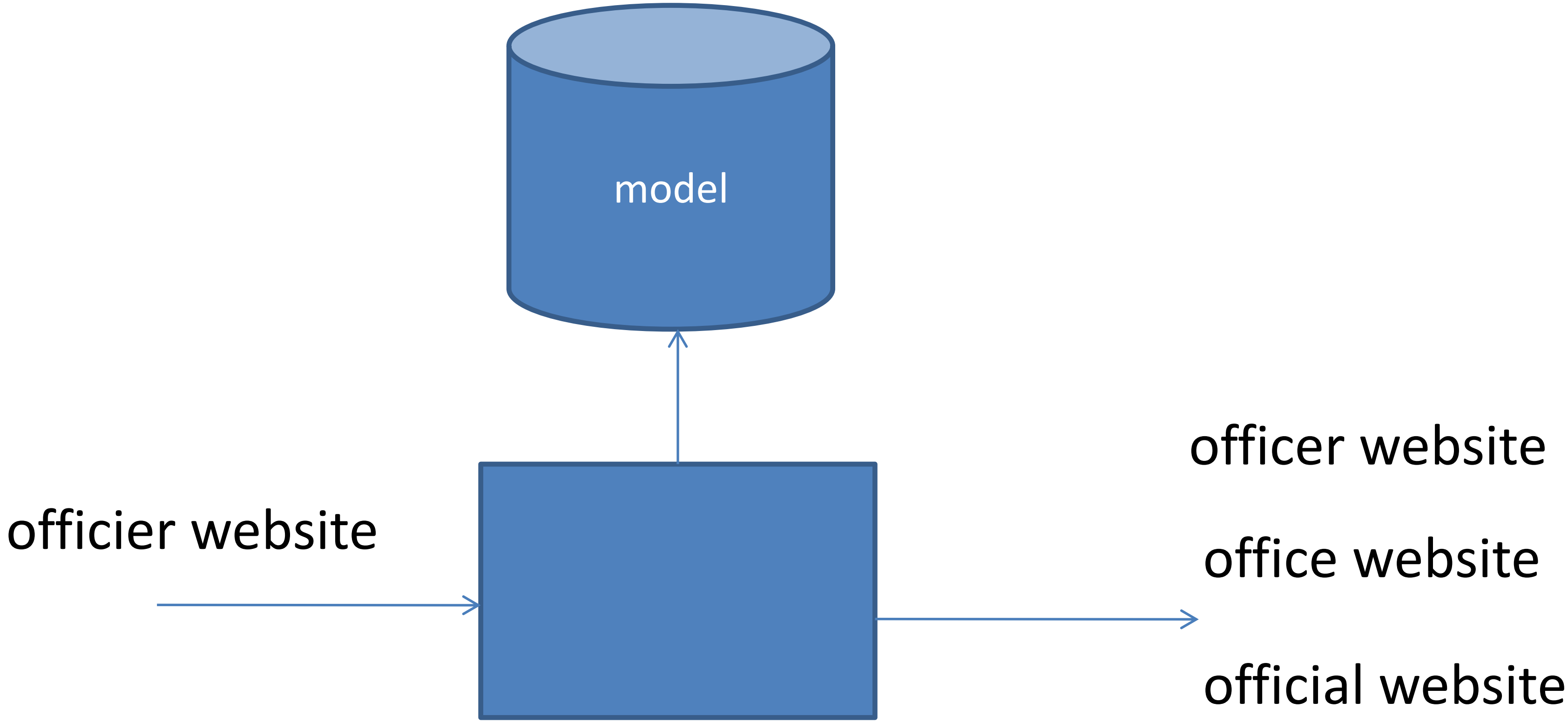
Jiafeng Guo, Gu Xu, Hang Li, Xueqi Cheng

SIGIR 2008

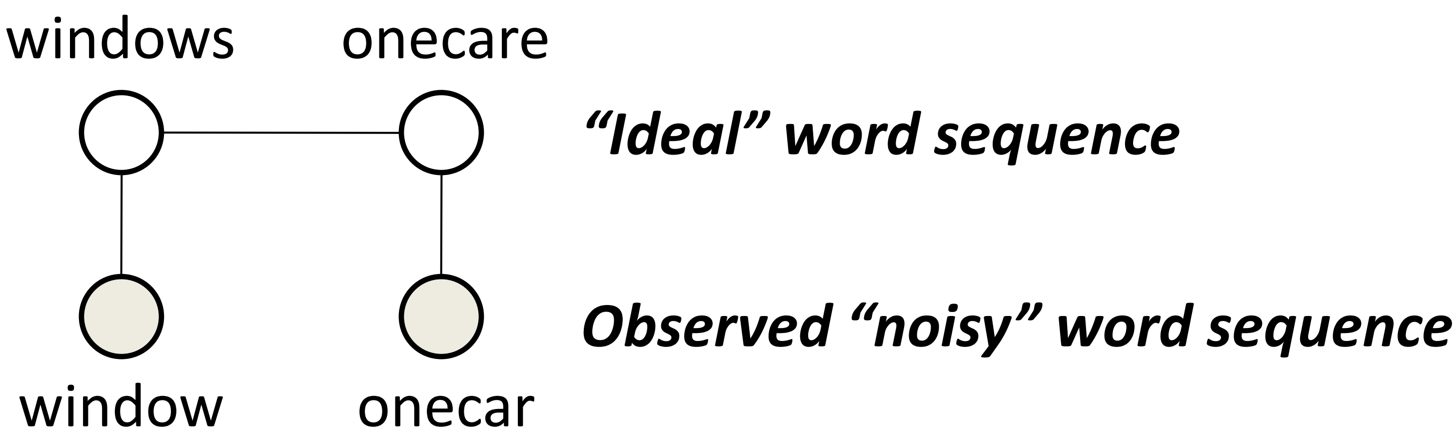
Candidate Selection in Spelling Error Correction Our Approach = Conditional Random Fields



Candidate Selection in Spelling Error Correction



Candidate Selection Problem



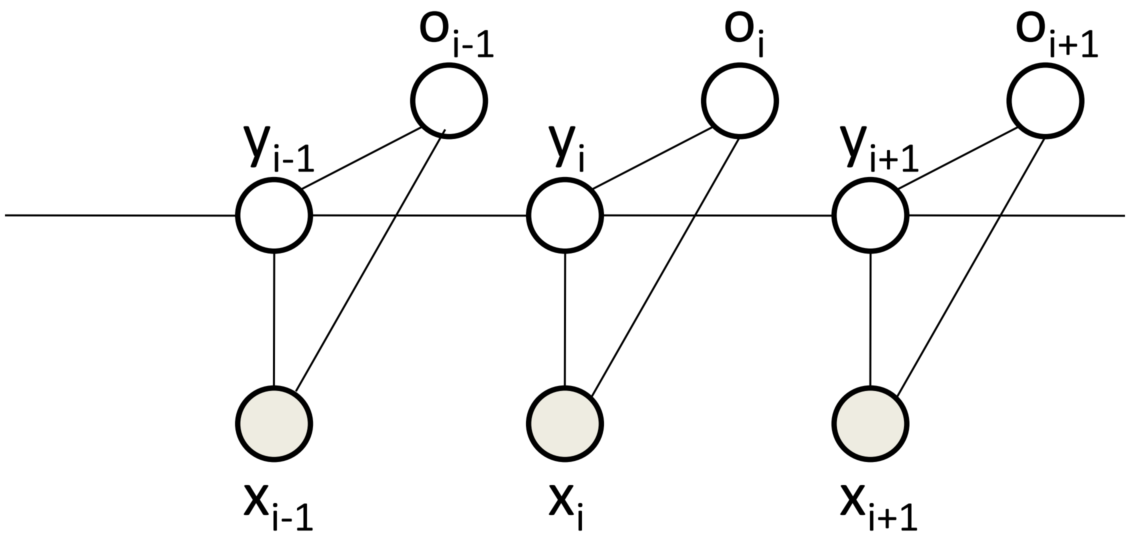
$$y^* = \arg\max_y \Pr(y|x)$$

"ideal" query word sequence

original query word sequence

Conditional Random Fields for Candidate Selection

Introducing Refinement Operations



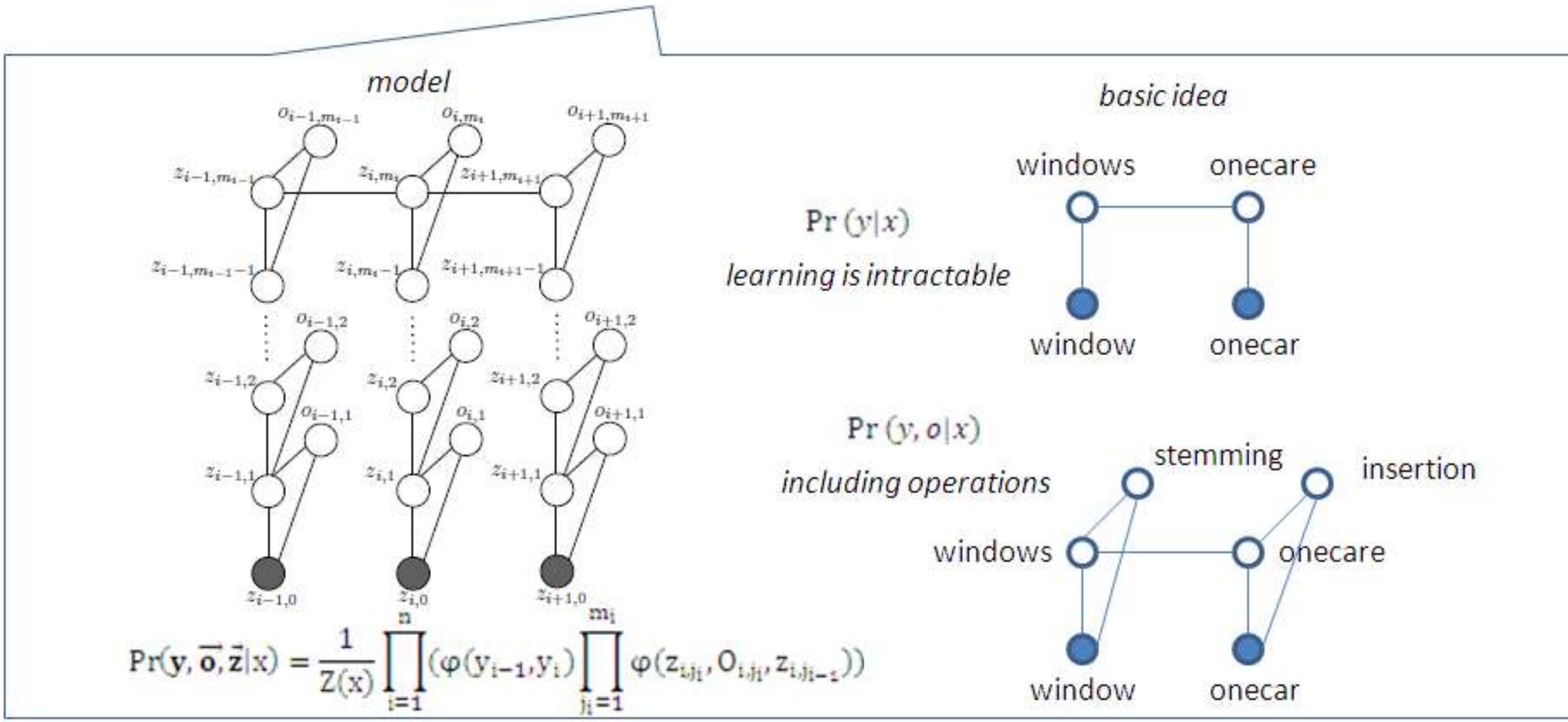
$$\Pr(y, o|x) = \frac{1}{Z(x)} \prod_{i=1}^{n} \phi(y_{i-1}, y_i)\phi(y_i, o_i, x)$$

Operations

Spelling: insertion, deletion, substitution, transposition, …

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

Query Refinement Using Conditional Random Fields



IR Matching (Relevance) Models

	Probabilistic Approach	**Non Probabilistic Approach**
Term Matching (unigram)	BM25[Robertson], LM4IR [Zhai][Ponte & Croft]	Vector Space Model [Salton]
Term Matching (n-gram)	MRF[Metzler & Croft]	*Similarity Function [Xu & Li]*
Topic Matching	PLSI[Hoffman], LDA[Blei et al]	LSI[Deerwester et al], *RLSI[Xu et al]*

IR Matching (Relevance) Models

	Probabilistic Approach	**Non Probabilistic Approach**
Sense Matching (synonym)		Rocchio [Rocchio], *Kernel Method [Wu et al]*
Sense Matching (spelling)	Generative model [Brill & Moore], *Log linear model [Wang et al], CRF [Guo et al]*	
Structure Matching	Translation Model [Berger & Lafferty]	

Summary

Summary

- Introduction to Web Search
- Relevance Model (Matching Model)
- Query Term Mismatch
- Learning to Match
- Our Methods
  - Robust Similarity Function Learning Using Kernel Methods
  - Regularized Latent Semantic Indexing
  - Query Generation Using Log Linear Model
  - Query Rewriting Using Conditional Random Fields

Thank You!

hangli@microsoft.com