Search and Discovery for Big Data

Xueqi Cheng

Institute of Computing Technology, CAS

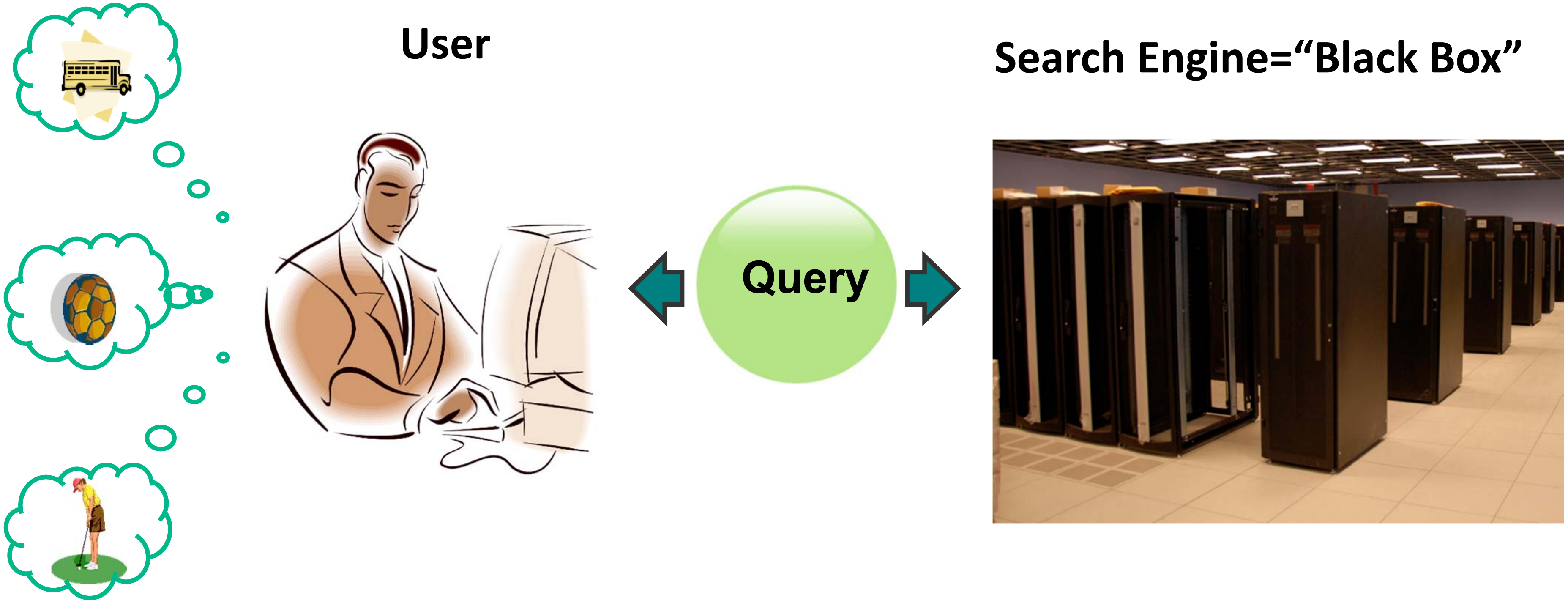
Outline

- Part I
  - Query understanding and topic modeling
- Part II
  - Learning-to-rank
- Part III
  - Social Media Analytics

Part I

Query understanding and topic modeling

1. Query Understanding



The First Step of IR

User

It is never easy to formulate a proper query to find what he/she needs.

Word ambiguity   Lack of knowledge

Unclear search intent   Unfarmiliar with SE

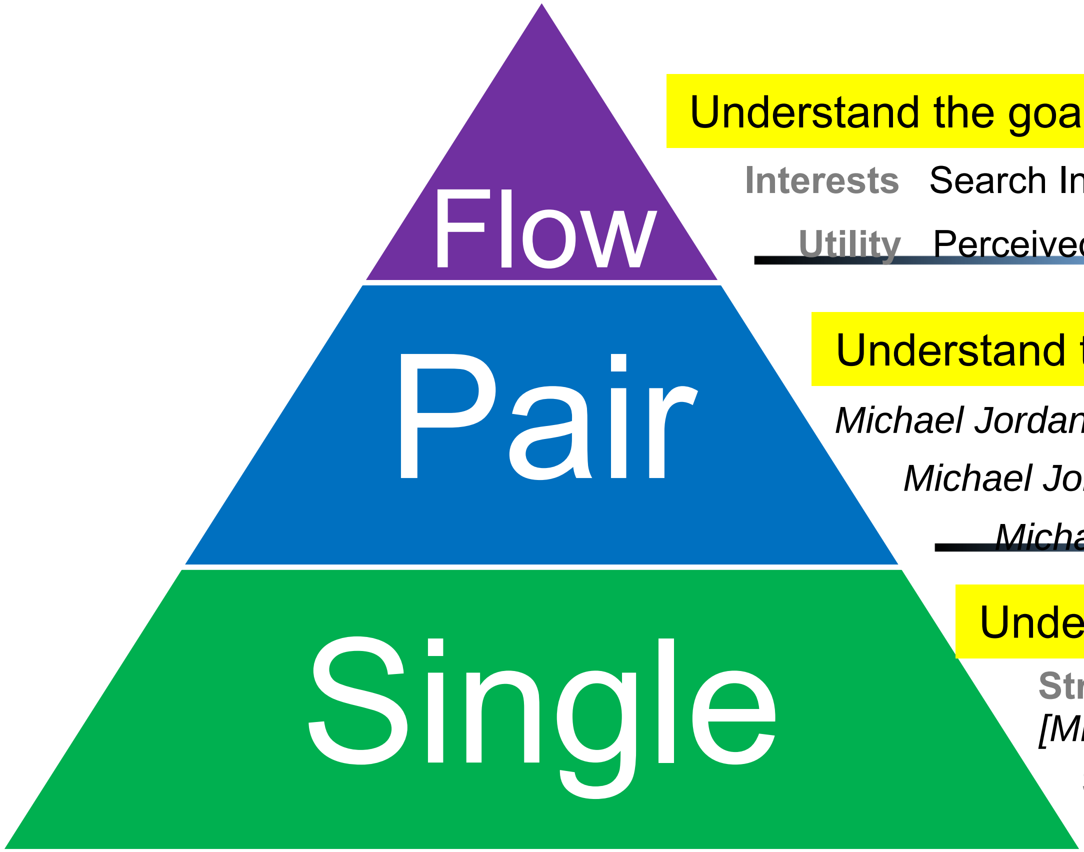
Search Engine

Understanding and representing users' search intent is critical for search success.

Short: lack of context   Ambiguous: multiple intents

Noisy: ill-formed

Different levels of Understanding



Understand the goal of the query

Interests Search Interests + Explortary Interests

Utility Perceived Utility + Posterior Utility

Understand the relation between queries

Michael Jordan ~ Michael Jordan Berkeley Intent: academic researcher

Michael Jordan ~ NBA Michael Jordan Intent: NBA star

Michael Jordan Berkeley ✕ *NBA Michael Jordan*

Understanding the representation of query

Structure

*[Michael Jordan: **PersonName**] [Berkeley: **Location**]*

Spelling

M Jordan Berkele → *Michael Jordan Berkeley*

Q: *M Jordan Berkele*

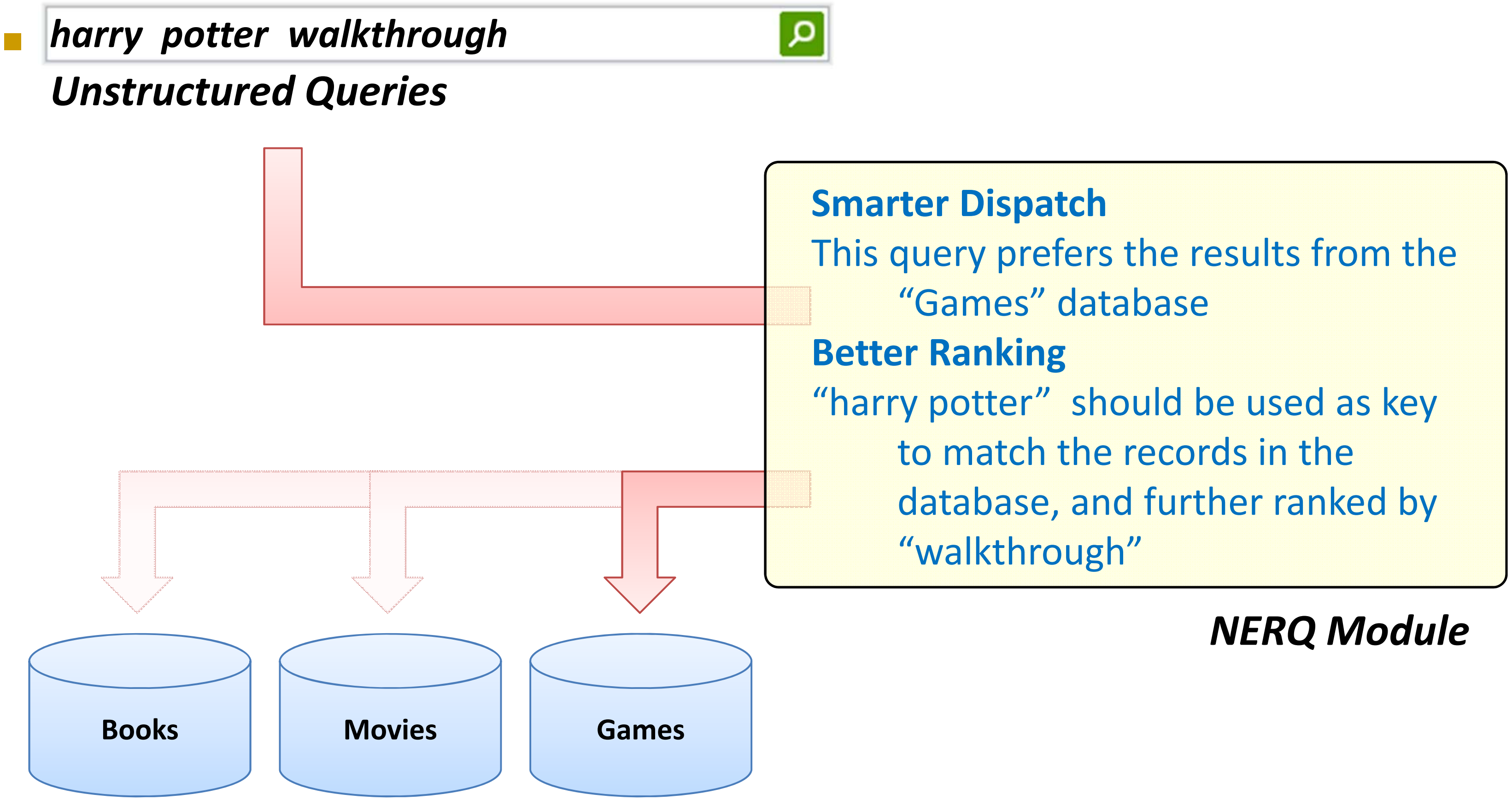
Understanding the Representation : Named Entity Recognition in Query （SIGIR'09）

Problem Definition

Named Entity Recognition in Query (NERQ)

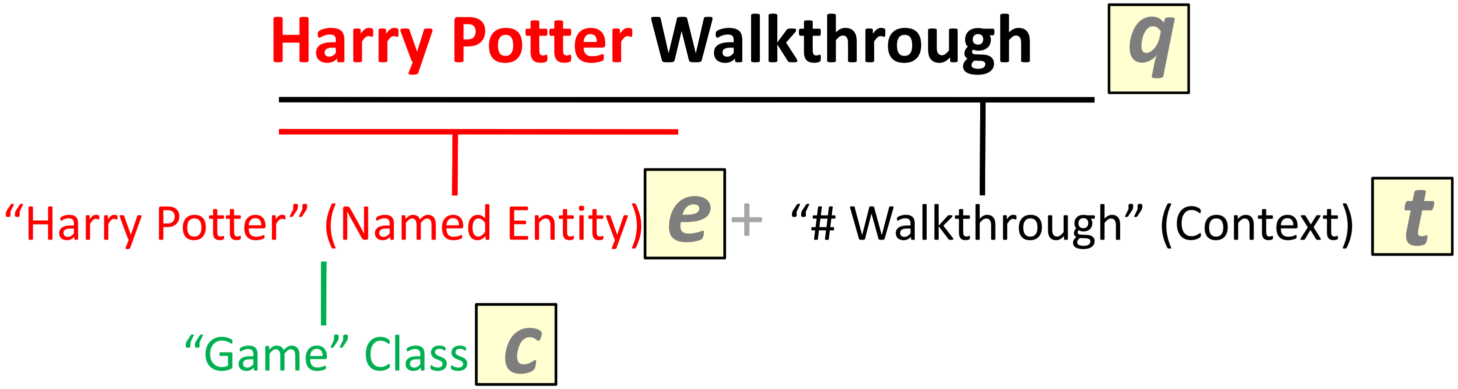
Identify Named Entities in Query and Assign them into Predefined Categories with Probabilities

NERQ in Searching Structured Data



Structured Databases
(Instant Answers, Local Search Index, Advertisements and etc)

Our Approach to NERQ



- Goal of NERQ becomes to find the best triple (e, t, c)* for query q satisfying

$$(\mathrm{e}, \mathrm{t}, \mathrm{c})^* = \arg\max_{(e,t,c)\in G(q)} p(e,t,c)$$
$$= \arg\max_{(e,t,c)\in G(q)} p(e)p(c|e)p(t|c)$$

Training with Topic Model using Query Log

- Training data $T = \{(e_i,\ t_i,\ *)\}$: Collected from Query Logs

$$\max \prod_{e} p(e) \prod_{i|e_i=e} \sum_{c} p(c|e)\, p(t_i|c)$$

e   ***t*** (Pseudo Doc)   ***c*** (Topics)

harry potter

\#	1000
\# movie	800
\# book	650
\# cheats	300
\# pics	200
\# summary	100

Movie, Book, Game

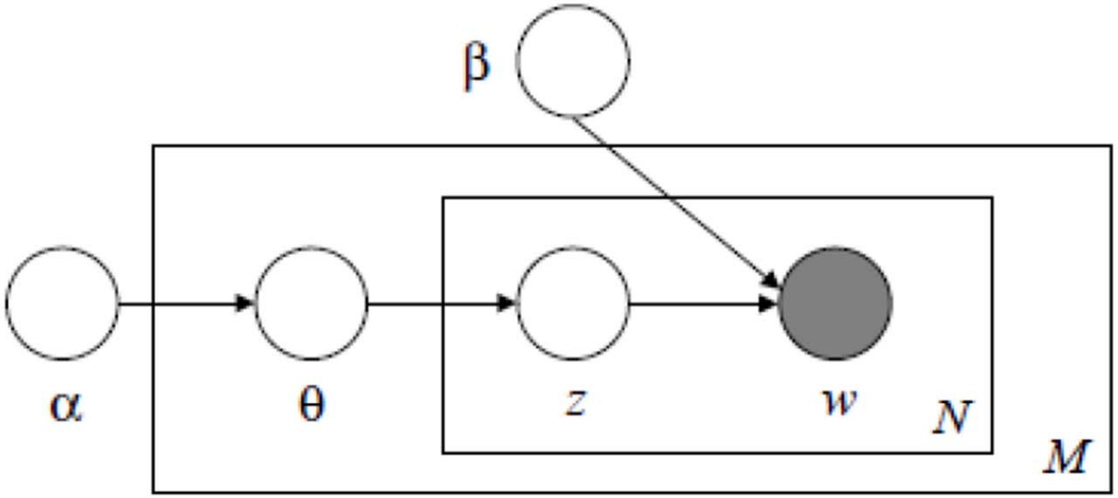
final fantasy

\#	300
\# movie	120
\# wallpaper	50
\# xbox	10
\# soundtrack	10

Movie, Game

 # is a placeholder for name entity

Latent Dirichlet Allocation



z: *Movie, Book, Game*

w: \#, \# movie, \# book, ….

θ : distribution of classes for named entity

β : distribution of contexts for class

Weakly Supervised LDA(WS-LDA)

- LDA + Soft Constraints (w.r.t. Weak Supervisions)
  - Align latent topics to predefined classes

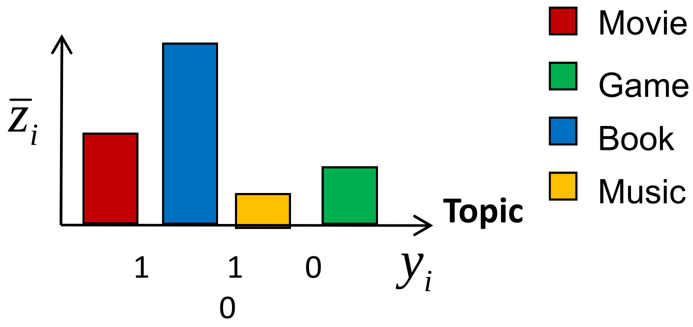
$$O(w, y) = \underbrace{\log p(w|\alpha, \beta)}_{\textit{LDA Probability}} + \underbrace{\lambda C(y|\alpha, \beta)}_{\textit{Soft Constraints}}$$

- Soft Constraints

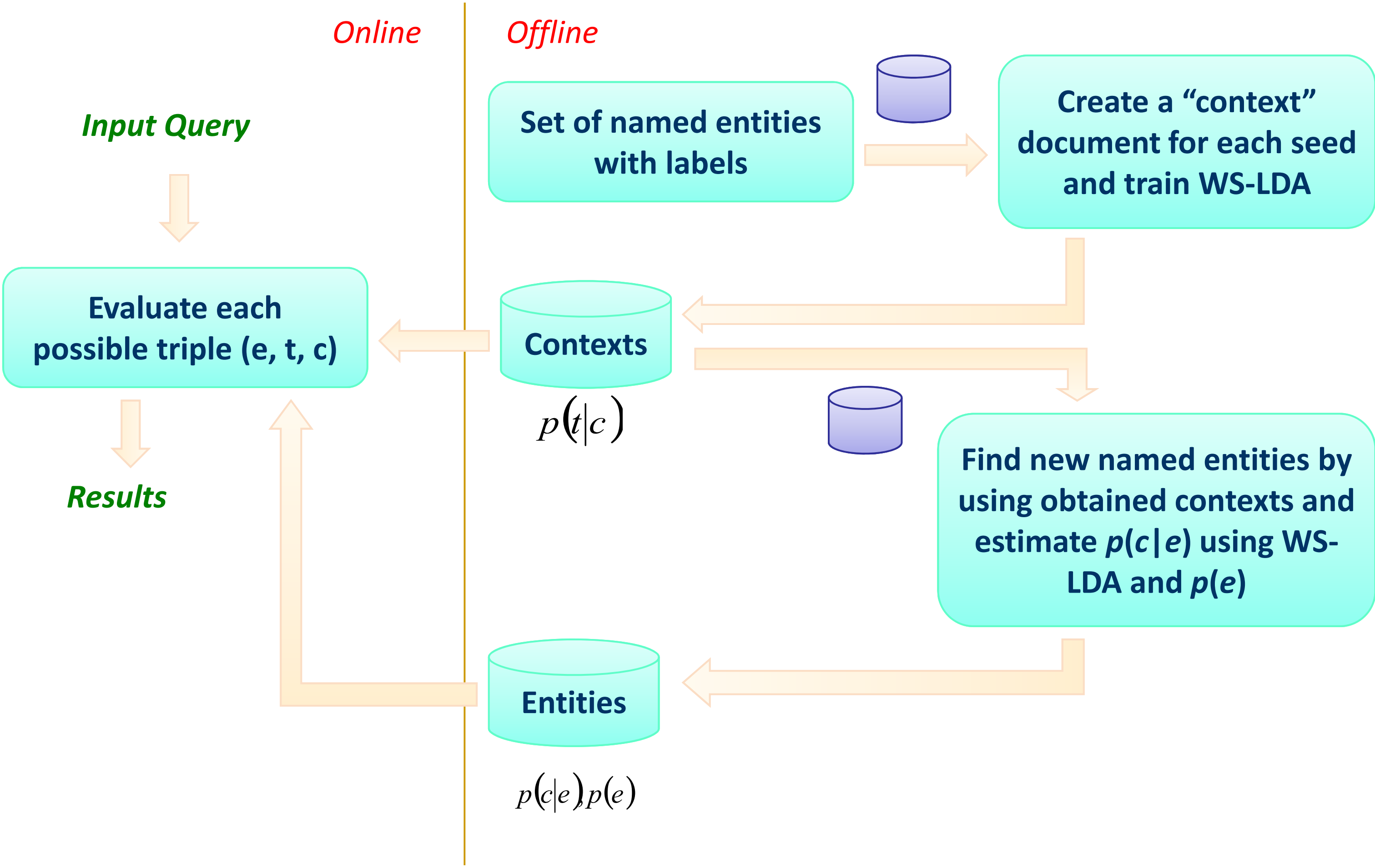
$$C(y|\alpha, \beta) = \sum_i \boxed{y_i}\,\boxed{\bar{z}_i}$$

Document Probability on the i-th Class

Document Binary Label on the i-th Class



System Flow Chat



Result

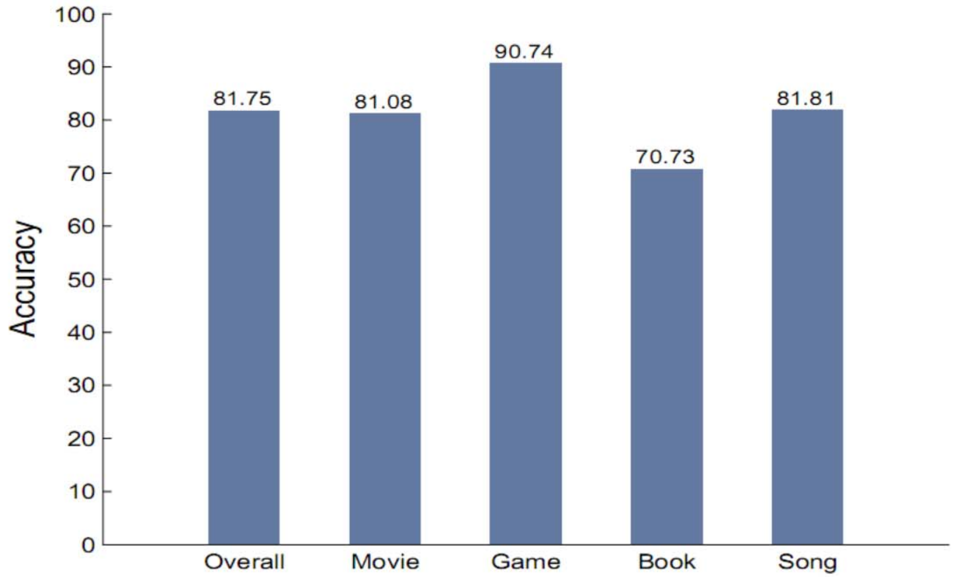


Table 6: Comparisons on Ranked Candidate Named Entities of each Class (P@N)

	Movie		Game		Book		Music		Average-Class	
	Determ	WS-LDA	Determ	WS-LDA	Determ	WS-LDA	Determ	WS-LDA	Determ	WS-LDA
P@25	0.92	1	0.98	1	0.84	1	0.96	1	0.92	1
P@50	0.9	1	0.96	1	0.82	1	0.92	1	0.905	1
P@100	0.85	1	0.93	0.98	0.79	0.98	0.89	1	0.865	0.99
P@150	0.82	1	0.92	0.953	0.767	0.98	0.833	1	0.835	0.983
P@250	0.724	0.988	0.896	0.928	0.732	0.968	0.76	0.984	0.778	0.967

Understanding the Relation: Intent-Aware Query Similarity （CIKM’11）

Motivation

Apple

Apple tree

search intent:
looking for apple fruits

Apple store

search intent:
find products of the apple company

Intent-aware query similarity

Similarity between queries defined upon search intent

Existing Methods

	Intent-Not-Aware	Intent-Aware
Pare-wise Measures Independent measured on each pair Jaccard coefficient [Beeferman et al. 2000] cosine similarity [Baeza-Yates et al. 2004; Wen et al. 2002] Hybrid methods [Zhang et al. 2006; Jones et al. 2006] Jaccard & cosine [Deng et al. 2009] Kernel method [Sahami et al. 2006]	Problem: Mixed representation Biased by popular intent Ignore unpopular ones Apple ~Apple store Apple $\nsim$Apple tree ✗	✓
Graph-based Measures Propagate similarity over query relation graph Random walk [Craswell et al. 2007] hitting time [Mei et al. 2008] SimRank [Antonellis et al. 2008] Matrix Factorization [Ma et al. 2008] Graph Projection [Bordino et al. 2010]	Problem: Propagate across the boundary Wrongly connect queries from different search intents Apple store ~ Apple tree ✗	✓

Overview

A. Identify the potential search intent of queries

B. Intent-aware similarity measure

I. Extract intent-aware representations

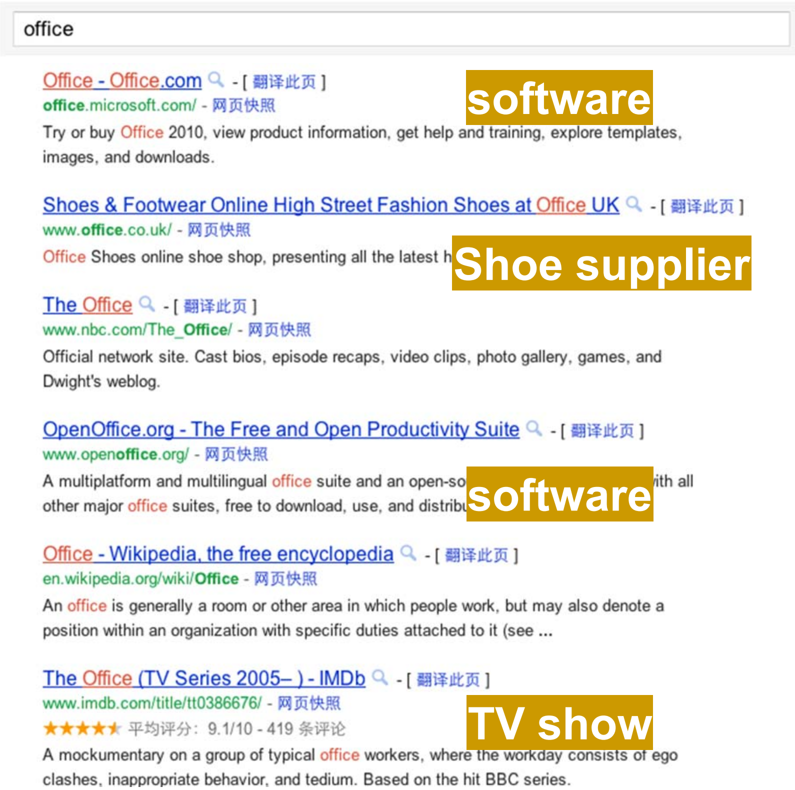
II. Apply different types of similarity measures

A. Identify Search Intents (Data)

leverage two types of auxiliary data

Search result snippets

Great Context Describing the query

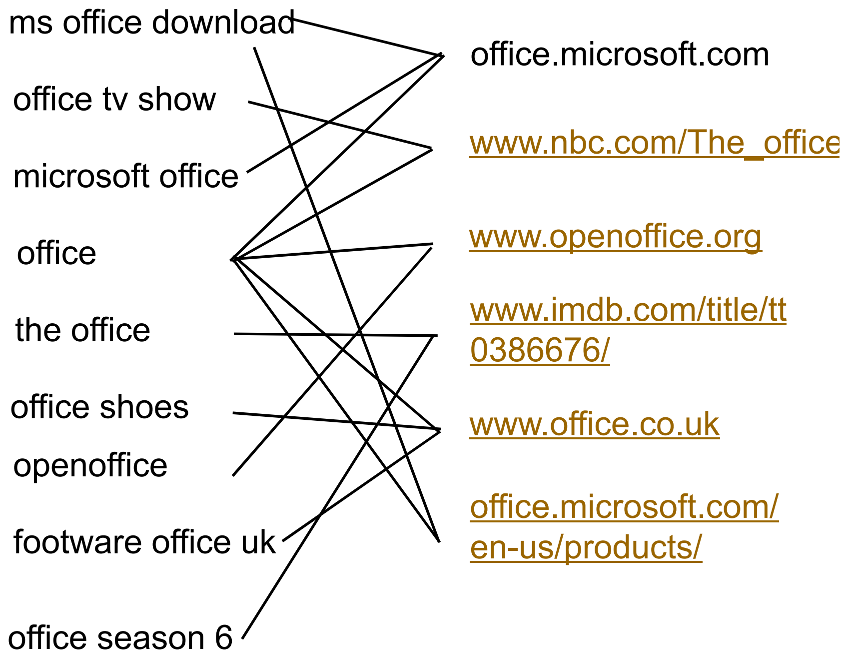


Pro: higher recall

Con: irrelevant/spam/advertisement/ambiguity

Clickthrough

Precise information from
Wisdom of crowds



Pro: higher precision

Con: sparse

A. Identify Search Intents (Algorithm)

Search result snippets

Topic Model

PLSI model

top search result snippets ⟶ virtual documents

words in snippets ⟶ words

potential search intents ⟶ topics

1. select a query q_i with probability $P(q_i)$,
2. pick a potential search intent s_k with probability $P(s_k|q_i)$
3. generate a word w_j with probability $P(w_j|s_k)$.

log-likelihood
$$\boxed{\tilde{\mathcal{L}} = \sum_{i=1}^{N}\sum_{j=1}^{M} n(q_i, w_j)\log\left(P(q_i)\sum_{k=1}^{K} P(w_j|s_k)P(s_k|q_i)\right)}$$

Clickthrough

Regularization

powerful constraint: two queries share many same clicked URLs ⟶ convey similar search intent

$$\boxed{\mathcal{R} = \sum_{i,j=1}^{N}\sum_{k=1}^{K} C_{ij}(P(s_k|q_i) - P(s_k|q_j))^2}$$

co-click matrix

Regularized Topic Model

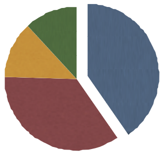
$$\begin{aligned}\mathfrak{L} &= \mathcal{L} - \lambda\mathcal{R} \\ &= \sum_{i=1}^{N}\sum_{j=1}^{M} n(q_i, w_j)\log\left(P(q_i)\sum_{k=1}^{K} P(w_j|s_k)P(s_k|q_i)\right) - \lambda\sum_{i,j=1}^{N}\sum_{k=1}^{K} C_{ij}(P(s_k|q_i) - P(s_k|q_j))^2\end{aligned}$$

B.Intent-Aware Similarity Measure (Pair-wise)

Similarity independently measured by pair-wise metrics

I. Extract intent-aware representations

original: word vector representation $\vec{q}_i[l] = n(q_i, w_l)$



intent-aware: word vector representation under k-th search intent $\vec{q}_{ik}[l] = n(q_i, w_l)\underline{P(s_k|q_i, w_l)}$

expected search intent distribution for each word occurrence w_l given query q_i

II. Apply Pair-wise similarity measures

similarity under k-th search intent $Sim_k(q_i, q_j) = \dfrac{\vec{q}_{ik} \cdot \vec{q}_{jk}}{\| \vec{q}_{ik} \| \| \vec{q}_{jk} \|}$

B.Intent-Aware Similarity Measure (Graph-based)

$\boxed{\text{similarity calculated over the query graph}}$

I. Extract intent-aware representations

original: query similarity graph adjacency matrix $A = [W_{ij}]_{i,j=1,\dots,N}$

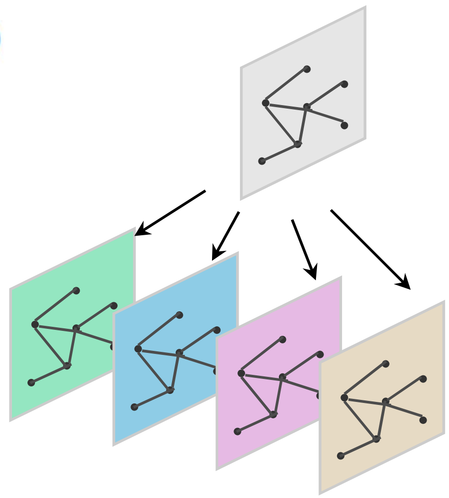
Jaccard coefficient

intent-aware: the probability that an edge will be generated between query q_i with search intent s_k and query q_j with search intent s_l $P(s_k|q_i)P(s_l|q_j)$

$\sum_{k,k'} P(s_k|q_i)P(s_l|q_j) = 1$

query similarity graph under k-th search intent $W_{ij}^k = W_{ij}P(s_k|q_i)P(s_k|q_j)$

II. Apply Graph-based similarity measures



spectral embedding $L_k\mathbf{y} = \lambda D_k\mathbf{y}$

query representation under k-th search intent $\vec{q}_{ik} = (\mathbf{y}_1(i), \dots, \mathbf{y}_m(i))$

similarity under k-th search intent $Sim_k(q_i, q_j) = \dfrac{1 + cos(\vec{q}_{ik}, \vec{q}_{jk})}{2}$

Result

Table 1: Example Queries Pairs with Similarity Scores Calculated by Different Methods

Method	Intent†	apple					
		apple store	apple company	apple ipod	apple fruit	apple tree	apple juice
Cos-Word	N/A	0.86	0.78	0.65	0.17	0.15	0.11
Cos-Intent	fruit	0	0	0	0.44	0.41	0.39
	company	0.92	0.83	0.77	0	0	0
Embed-Click	N/A	0.89	0.81	0.87	0.46	0.37	0.41
Embed-Intent	fruit	0	0	0	0.83	0.77	0.79
	company	1	0.96	0.99	0	0	0

Method	Intent†	taylor					
		taylor swift	taylor swift new songs	taylor ice cream	taylor soft serve machine	taylor acoustic	taylor guitars
Cos-Word	N/A	0.55	0.51	0.49	0.58	0.62	0.59
Cos-Intent	singer	0.76	0.68	0	0	0	0
	instrument	0	0	0	0	0.87	0.85
	company	0	0	0.52	0.61	0	0
Embed-Click	N/A	0.48	0.47	0.47	0.46	0.44	0.51
Embed-Intent	singer	1	1	0	0	0	0
	instrument	0	0	0	0	0.60	0.63
	company	0	0	0.87	0.72	0	0

†the search intents are manually labeled for illustration

Examples of Similar and Dissimilar Query Pairs

Type	Query Pair	Traditional Method		Intent-Aware Method†	
		Cos-Word	**Embed-Click**	**Cos-Intent**	**Embed-Intent**
Similar Pairs	(apple, apple store)	0.86	0.89	0|0.92	0|1
	(apple, apple fruit)	0.17	0.46	0.44|0	0.83|0
Dissimilar Pairs	(apple store, apple fruit)	0.09	0.37	0|0	0|0
	(apple ipod, apple tree)	0.08	0.34	0|0	0|0
Similar Pairs	(taylor, taylor swift)	0.55	0.48	0.76|0|0	1|0|0
	(taylor, taylor soft serve machine)	0.58	0.46	0|0|0.61	0|0|0.72
Dissimilar Pairs	(taylor swift, taylor soft serve machine)	0.28	0.36	0|0|0	0|0|0
	(taylor ice cream, taylor acoustic)	0.24	0.38	0|0|0	0|0|0

†similarity scores under different intents are separated by vertical bars for clarity

Result

Table 3: Examples of Manually Built Test Set

Query	Major Intents
24 hours	1. tv show 24, 24 on fox, 24 the series 2. 24 fitness, 24hr fitness, 24 hour gym
sigma	1. sigma aldrich, sigma chemicals, sigma biology 2. greek alphabet sigma, sigma symbol, sigma maths 3. sigma camera, sigma photo, sigma lenses
svm	1. svm cards, svm gift card, svm gas cards 2. svm kernel, svm tutorial, support vector machine

Expected Inter-intent Similarity:

$$InterSim(S)=\frac{1}{K(K-1)}\sum_{S_k,S_{k'}\in S,k\neq k'}\left[\sum_{q_i\in S_k}\sum_{q_j\in S_{k'}}\frac{Sim(q_i,q_j)}{|S_k||S_{k'}|}\right]$$

Expected Intra-intent Similarity:

$$IntraSim(S)=\frac{1}{K}\sum_{k=1}^{K}\left[\sum_{q_i,q_j\in S_k,i\neq j}\frac{2Sim(q_i,q_j)}{|S_k||S_k-1|}\right]$$

Expected inter-intra ratio $$\mathcal{H}_{\hat{S}}(Sim)=E\left[\frac{InterSim(S)}{IntraSim(S)}\right]_{S\in\hat{S}}$$

$\mathcal{H}_{\hat{S}}(Sim)$ for Different Similarity Measures

Method	$\mathcal{H}_{\hat{S}}(Sim)$	Significant differences†
Cos-Word	0.47±0.06	>Embed-Click***
Cos-Intent	0.08±0.03	>Cos-Word*** >Embed-Click***
Embed-Click	0.54±0.02	
Embed-Intent	0.09±0.03	>Cos-Word*** >Embed-Click***

†the significant levels are denoted as 0.1* 0.05 ** 0.01 ***

Understanding the Goal:

More Than Relevance: High Utility Query Recommendation By Mining Users' Search Behaviors(CIKM'12, ECIR'13)

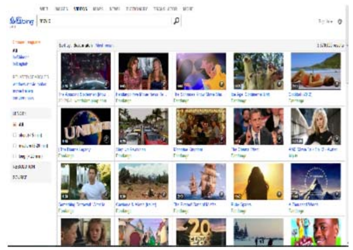
Motivation

Information Seeking Tasks

Query

not easy to formulate properly

Find Web pages



Locate resources

Access Info of topics

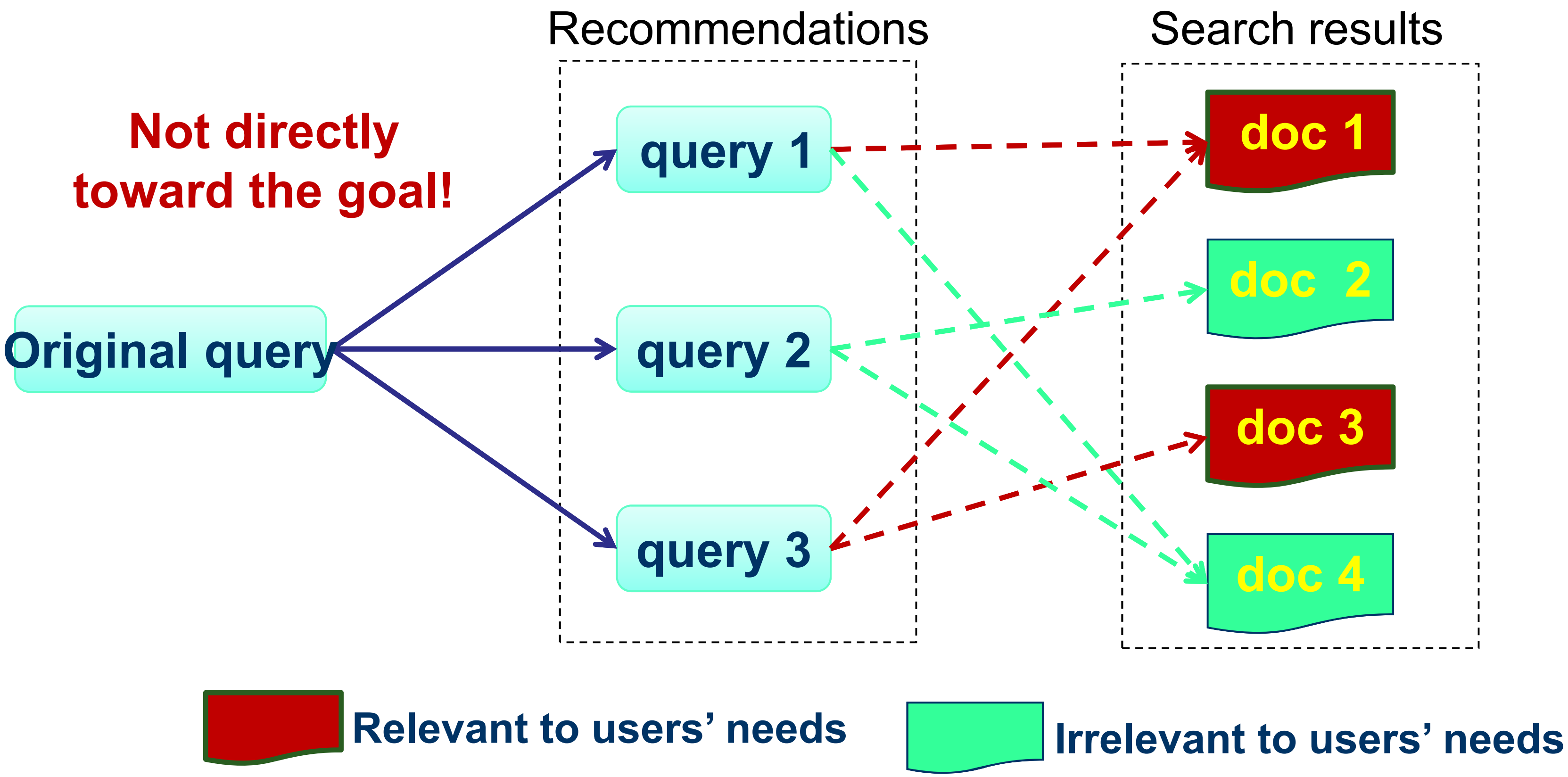
The ultimate goal of query recommendation

Assist users to reformulate queries so that they can **acquire their desired information successfully and quickly**

Motivation

Relevant query recommendation:
Providing alternative queries similar to a user's initial query

Problem:
relevant query $\not\Rightarrow$ satisfy users' needs **not necessarily**

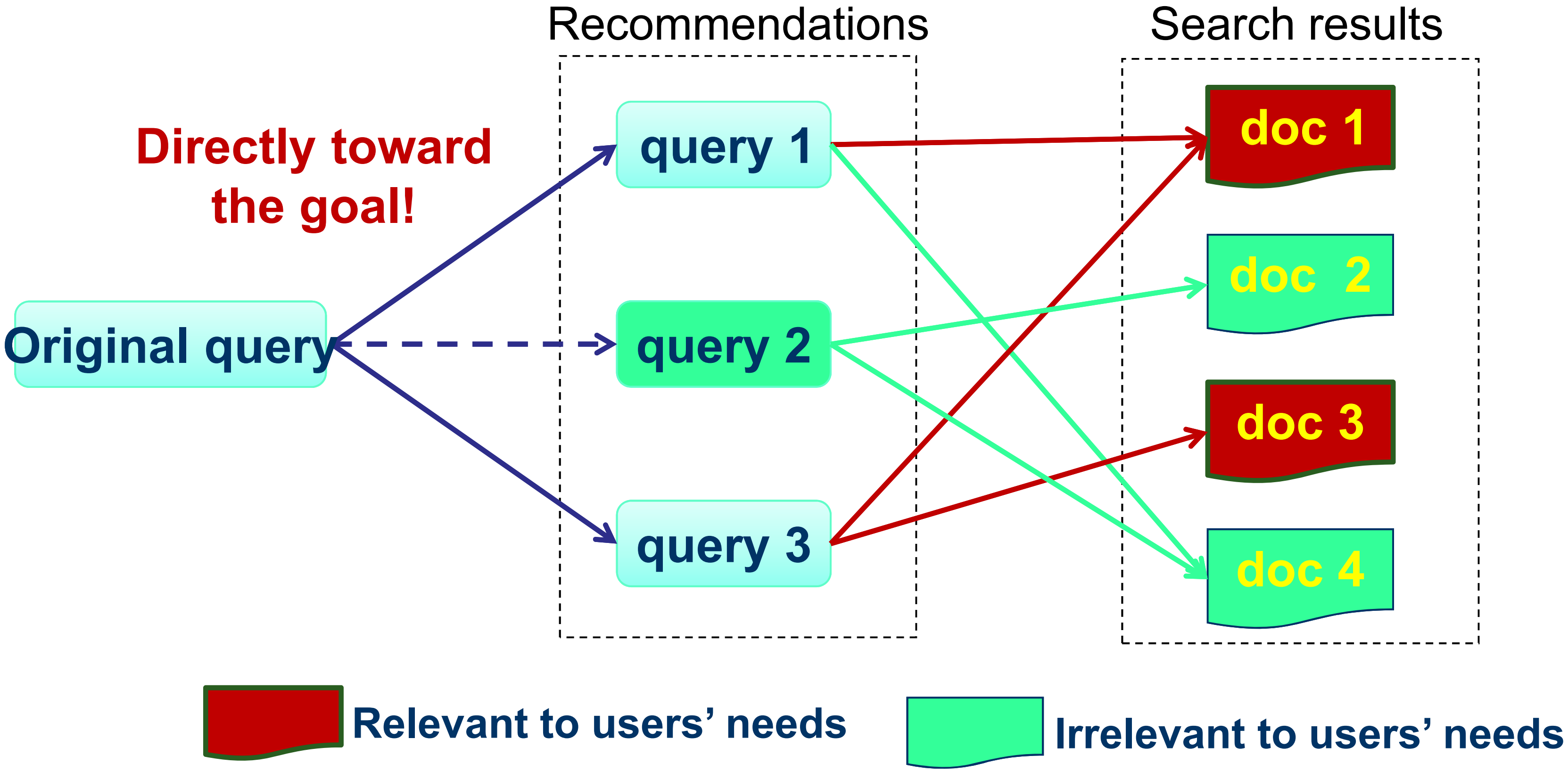


Motivation

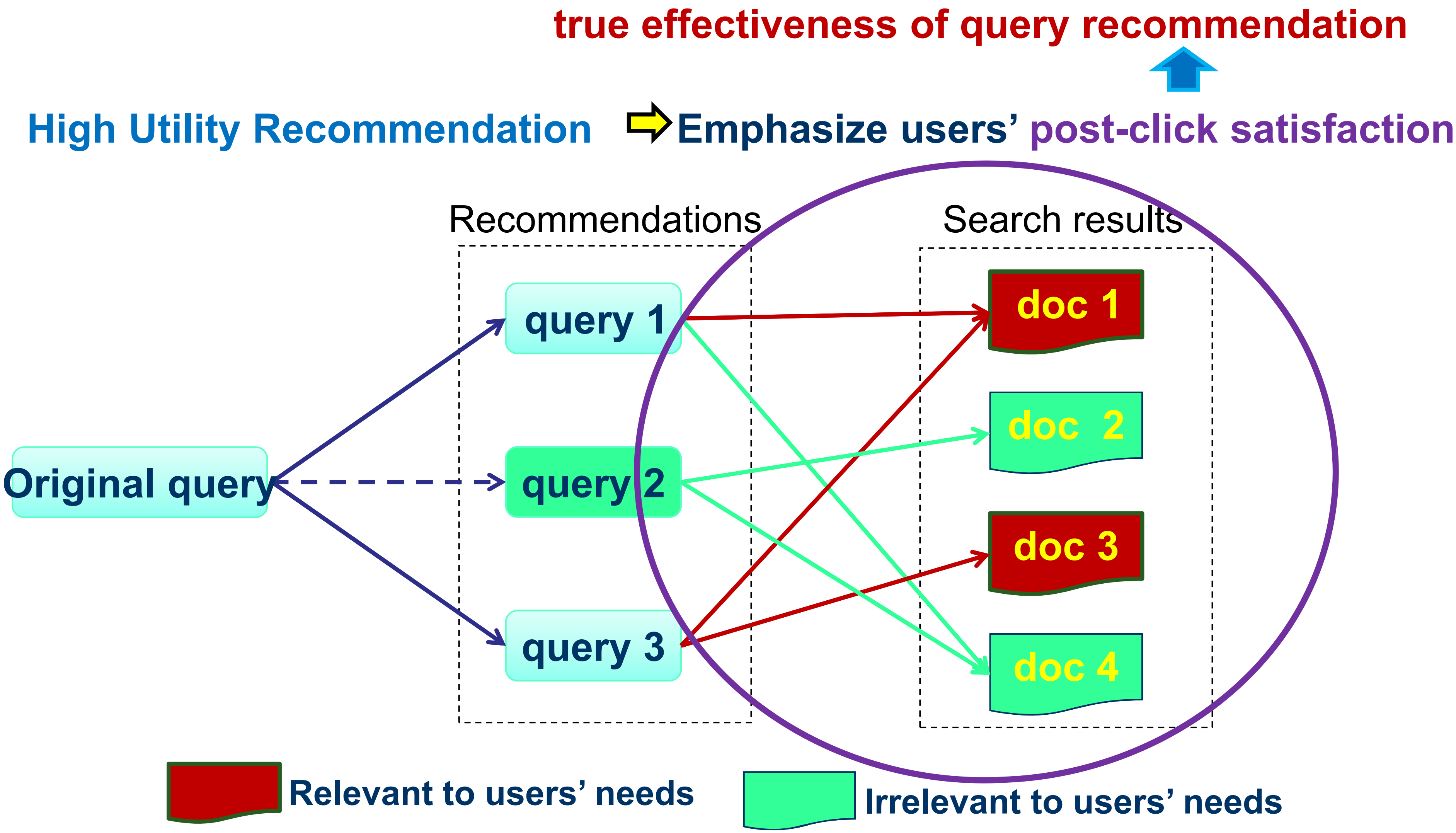
High Utility Recommendation:
Providing queries that can better satisfy users' information needs

Query Utility Definition:

The **information gain** that a user can obtain from the search results of the query according to her original information needs.



Motivation



true effectiveness of query recommendation
High Utility Recommendation
Emphasize users' post-click satisfaction
Recommendations
Search results
Original query
query 1
query 2
query 3
doc 1
doc 2
doc 3
doc 4
Relevant to users' needs
Irrelevant to users' needs

Challenges for high utility recommendation

→ How to infer query utility?

Query Utility Model

→ How to evaluate?

Two evaluation metrics

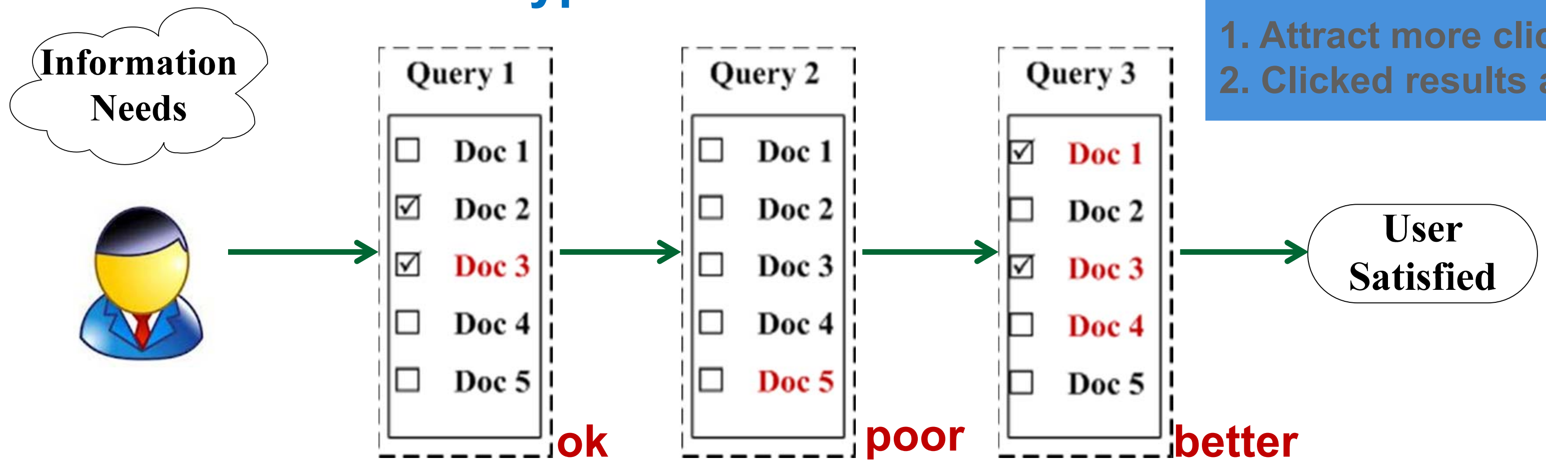
Our Approach

how to infer query utility?

Key Idea: Through user's search behaviors

A typical search session

1. Attract more clicks
2. Clicked results are relevant



Red - relevant √ - clicked

Perceived Utility

model the attractiveness of the search results

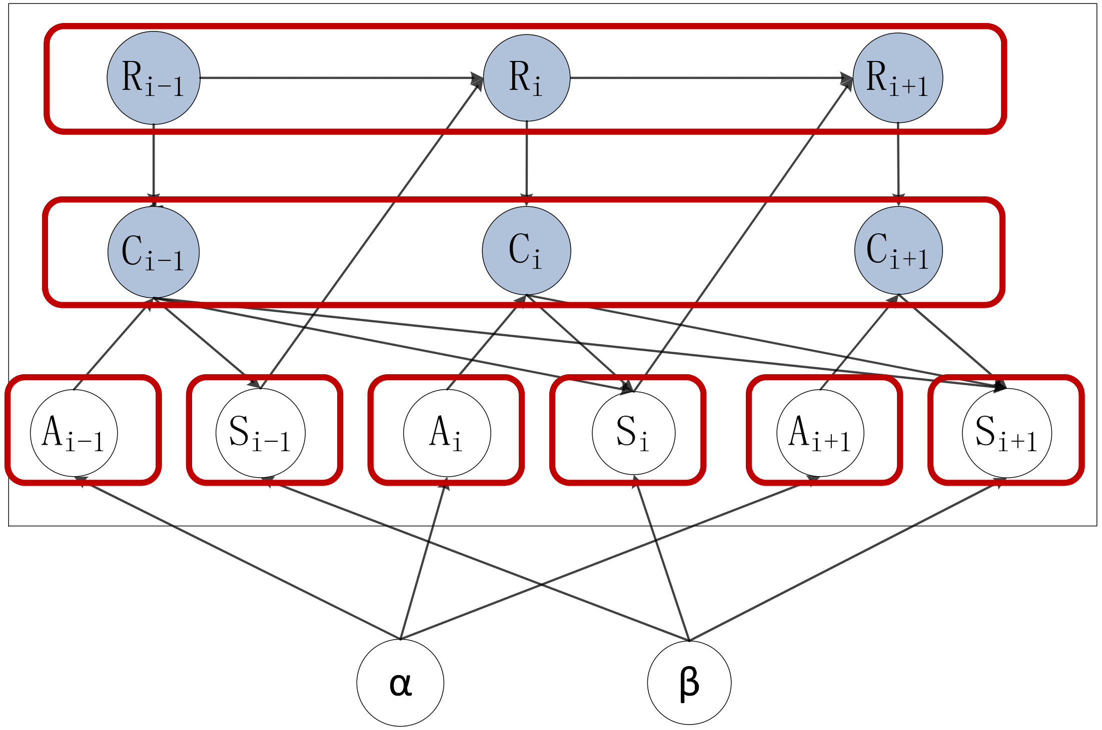
Query Utility

Posterior Utility

model the satisfaction of the clicked search results

Query Utility Model (dynamic Bayesian network)

how to infer query utility?



$$P(R_i = 1 | R_{i-1} = 1, S_{i-1} = 1) = 0.$$

$$P(C_i = 1 | R_i = 1, A_i = 1) = 1,$$

$$P(A_i = 1) = \alpha_{\phi(i)},$$

$$P(S_i = 1 | C_{1:i}) = \sigma(\sum_{k=1} \beta_{\phi(k)} \cdot I(C_k{=}1)),$$

$$\sigma(x) = \frac{1}{1+e^{-x}}.$$

Perceived Utility α : control the probability of the attractiveness

Posterior Utility β : control the probability of users' satisfaction

R_i: whether there is a reformulation at position i

C_i: whether the user clicks on sor **Query Utility $\mu_t = \alpha_t * \beta_t$** ulation at position i;

A_i: whether the user is attracted by the search results of the reformulation at position I;

The expected information gain users obtained from the search results of the query according to their original information needs

Evaluation

how to evaluate?

Query Level Judgment

Original query

Recommendations

query 1 Relevant or Not?

query 2 Relevant or Not?

query 3 Relevant or Not?

Relevant = 1

Partial Relevant = 0.5

Irrelevant = 0

Evaluation

how to evaluate?

Document Level Judgment

Original query

Recommendations & Clickthrough

query 1

doc 1 doc 2 doc 3

Relevant or Not?

query 2

doc 1 doc 2 doc 3

Relevant or Not?

query 3

doc 1 doc 2 doc 3

Relevant or Not?

UFindIt log data: http://ir-ub.mathcs.emory.edu/uFindIt/ (SIGIR'11 Best Pa

Evaluation

how to evaluate?

– QRR (Query Relevant Ratio)

$$QRR(q) = \frac{RQ(q)}{N(q)}$$

Measuring the probability that a user finds(clicks) relevant results when she uses query q for her search task.

– MRD (Mean Relevant Document)

$$MRD(q) = \frac{RD(q)}{N(q)}$$

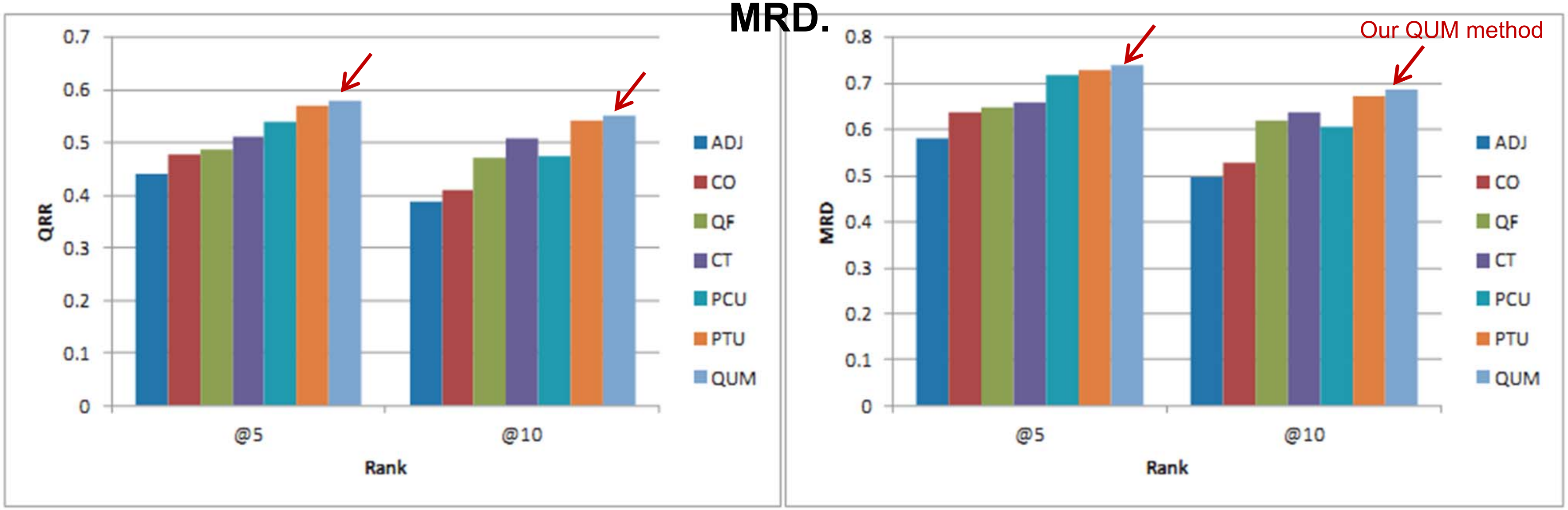
Measuring the average number of relevant results a user finds(clicks) when she uses query q for her search task.

Baseline Methods

- **Frequency-based methods**
  - Adjacency (ADJ) (WWW 06)
  - Co-occurrence (CO) (JASIST 03)
- **Graph-based methods**
  - Query-Flow Graph (QF) (CIKM 08)
  - Click-through Graph (CT) (CIKM 08)
- **Component utility methods**
  - Perceived Utility (PCU)
  - Posterior Utility (PTU)

Experimental Results

Comparison of the performance of all approaches (ADJ,CO,QF,CT,PCU,PTU,QUM) in terms of QRR and MRD.



(a) QRR (b) MRD

The performance improvements are significant
(t-test，p-value <= 0.05)

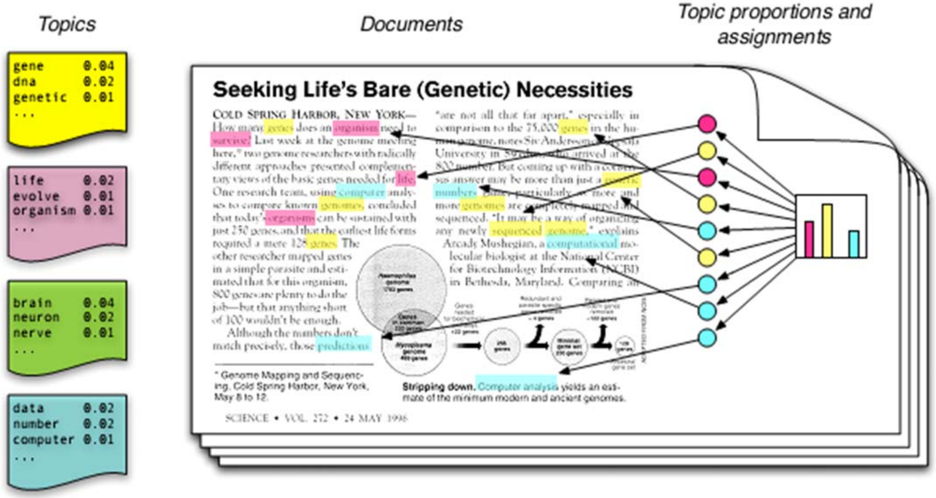
Experimental Results

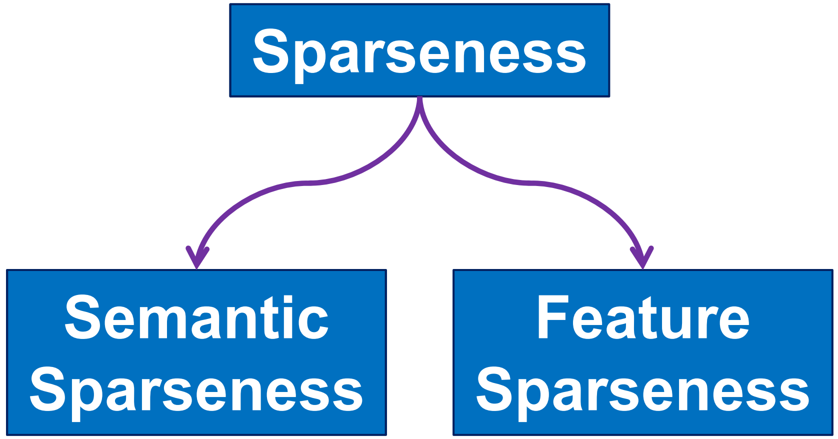
The improvement is larger on difficult queries!

Query Difficulty	Method	QRR @5	QRR @10	MRD @5	MRD @10
Easy	ADJ	0.588(18.64%)	0.526(26.30%)	0.771(20.32%)	0.674(25.22%)
	CO	0.609(14.55%)	0.529(25.63%)	0.830(11.80%)	0.687(22.89%)
	QF	0.618(12.94%)	0.604(9.89%)	0.846(9.67%)	0.806(4.69%)
	CT	0.654(6.62%)	0.635(4.65%)	0.836(11.02%)	0.805(4.79%)
	PCU	0.656(6.37%)	0.611(8.74%)	0.889(4.35%)	0.798(5.79%)
	PTU	0.689(1.22%)	0.663(0.17%)	0.908(2.18%)	0.837(0.86%)
	QUM	**0.698**	**0.664**	**0.928**	**0.844**
Medium	ADJ	0.460(30.00%)	0.429(33.19%)	0.596(24.14%)	0.527(33.76%)
	CO	0.495(20.81%)	0.441(29.65%)	0.640(15.72%)	0.550(28.10%)
	QF	0.511(17.07%)	0.500(14.39%)	0.615(20.43%)	0.630(11.79%)
	CT	0.534(12.07%)	0.549(4.02%)	0.689(7.54%)	0.692(1.81%)
	PCU	0.544(9.91%)	0.485(17.74%)	0.703(5.31%)	0.588(19.76%)
	PTU	0.581(2.87%)	0.557(2.70%)	0.722(2.53%)	0.689(2.18%)
	QUM	**0.598**	**0.572**	**0.740**	**0.704**
Hard	ADJ	0.259(65.27%)	0.216(91.19%)	0.351(54.37%)	0.284(77.27%)
	CO	0.314(36.29%)	0.261(58.17%)	0.412(31.63%)	0.340(48.00%)
	QF	0.324(32.08%)	0.312(32.20%)	0.441(22.94%)	0.414(21.78%)
	CT	0.334(28.08%)	0.343(20.17%)	0.437(24.15%)	0.424(18.85%)
	PCU	0.404(5.90%)	0.324(27.07%)	0.534(1.54%)	0.413(22.02%)
	PTU	0.426(0.28%)	0.402(2.51%)	0.526(3.18%)	0.485(3.92%)
	QUM	**0.427**	**0.412**	**0.542**	**0.504**

2. Topic Modeling

Topic Modeling → "Big Data"





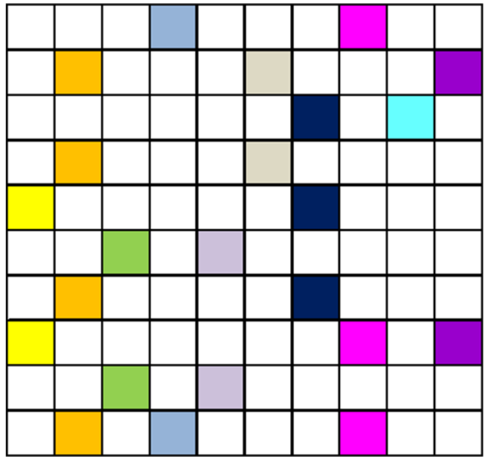
✓A generative probabilistic model
✓Documents are represented as random mixtures over latent topics
✓A topic is characterized by a distribution over words

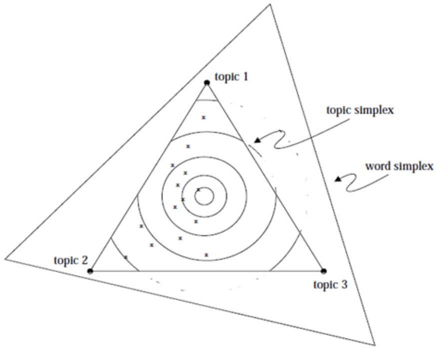
Topic Modeling (Semantic Sparseness):

Group Sparse Topical Coding: From Code to Topic (WSDM'13)

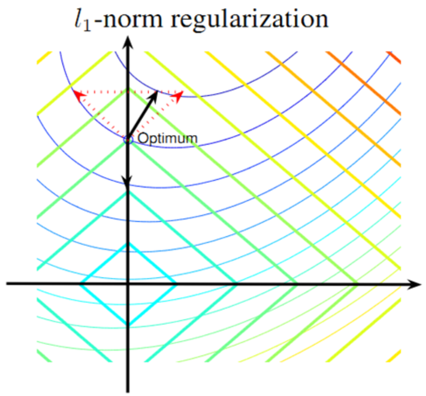
Sparse and Meaningful Topics

- Lots of data but relative sparse topics
- Traditional topic models
  - Probabilistic Model
    - Meaningful interpretation
    - Lack the control of sparsity
  - Non-probabilistic Model
    - Effective sparse controlling
    - Lack clear semantic meanings
- Can we enjoy the two merits?
  - Yes!!!





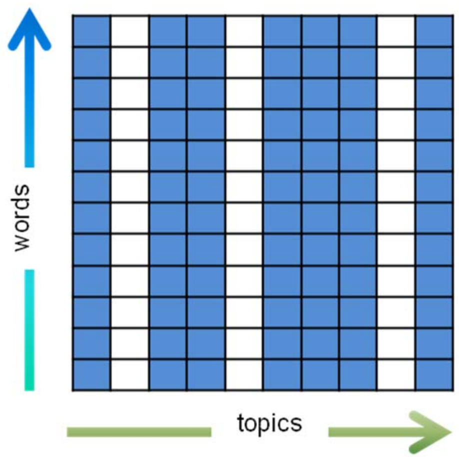
(Blei et al., ’03)



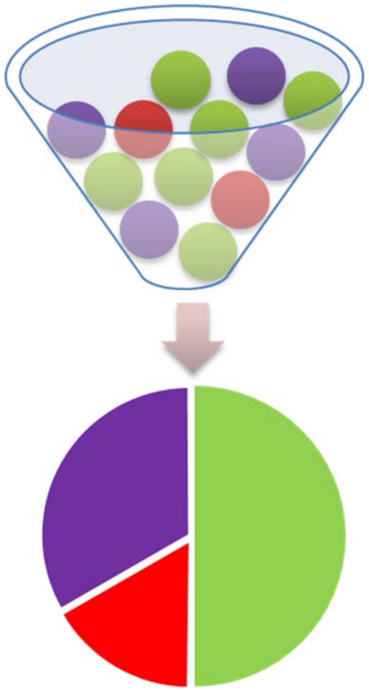
(Lin., ’08)

Basic Idea

- The meaning of document is composed of the meanings of words.
- Control the topics of words in turn control the topics of the document.



Word Topics



Doc Topics

Modeling the word count w_n in coding scheme

- To add sparse constraints

Codings of words are restricted by group lasso

- To align the sparse pattern of words' topics

Word count is generated from Poisson

- To recover the document 's topic proportion from code

Coding by STC

- Generative process:

1. For each topic $k \in \{1, \ldots, K\}$:

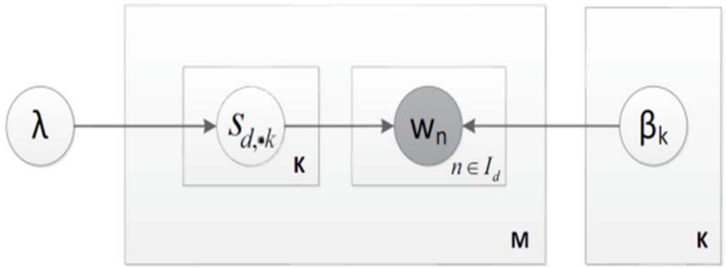
   Sample a word code vector $s_{\cdot k} \in \mathbb{R}^N \sim$ M-Laplace(λ).

2. For each observed word $n \in \boldsymbol{I}$:

   For each topic $k \in \{1, \ldots, K\}$:

   Sample a latent word count $w_{nk} \sim$ Poisson($s_{nk}\boldsymbol{\beta}_{kn}$).

3. Obtain the word count $w_n = \sum_{k=1}^{K} w_{nk}$.



- M-Laplace distribution: sparse codes (group-lasso in MAP-solution in log-space)
- Poisson distribution: additive property and Moran's property

Additive property

If $X_i \sim Poisson(\lambda_i), i = 1, \ldots, n$ are independent, and $\lambda = \sum_{i=1}^{n} \lambda_i$, then

$$Y = \sum_{i=1}^{n} X_i \sim Poisson(\lambda)$$

Coding by STC

- Joint distribution of word codes and word counts

$$\begin{aligned} p(s, w|\boldsymbol{\beta}) &= \prod_{k=1}^{K} p(s_{\cdot k}) \left(\prod_{n=1}^{|\boldsymbol{I}_d|} \prod_{k=1}^{K} p(w_{nk}|s_{nk}, \beta_{kn}) \right) \\ &= \prod_{k=1}^{K} p(s_{\cdot k}) \prod_{n=1}^{|\boldsymbol{I}_d|} p(w_n|s_{n\cdot}, \boldsymbol{\beta}), \end{aligned}$$

- Objective Function (MAP-estimation)

$$\begin{aligned} \min_{\Theta,\boldsymbol{\beta}} \mathcal{L}(\Theta, \boldsymbol{\beta}) &= -\ln P(\Theta, \boldsymbol{\beta}|D) \\ &= \min_{\Theta,\boldsymbol{\beta}} \sum_{d=1}^{M} \sum_{n=1}^{|\boldsymbol{I}_d|} \ell(s_{d,n\cdot}, \boldsymbol{\beta}) + \sum_{d=1}^{M} \sum_{k=1}^{K} \lambda \|s_{d,\cdot k}\|_2 + C \\ &= \min_{\Theta,\boldsymbol{\beta}} \sum_{d=1}^{M} \sum_{n=1}^{|\boldsymbol{I}_d|} \left(\sum_{k=1}^{K} s_{d,nk}\beta_{kn} - w_{d,n} \ln(\sum_{k=1}^{K} s_{d,nk}\beta_{kn}) \right) \\ &\quad + \sum_{d=1}^{M} \sum_{k=1}^{K} \lambda \|s_{d,\cdot k}\|_2 + C, \\ s.t. \quad & s_{d,n\cdot} \geq 0, \forall d, n \in \boldsymbol{I}_d, \\ & \sum_{n=1}^{N} \beta_{kn} = 1, \forall k, \end{aligned} \qquad (4)$$

Codes to Topics

- Topic proportion can be re-constructed from the word codes and dictionary

LEMMA 1. *[Moran's Property of Poisson Distribution] If variables $X_1, X_2, \cdots, X_n$ are independent Poisson random variables with parameters $\tau_1, \tau_2, \cdots, \tau_n$, then*

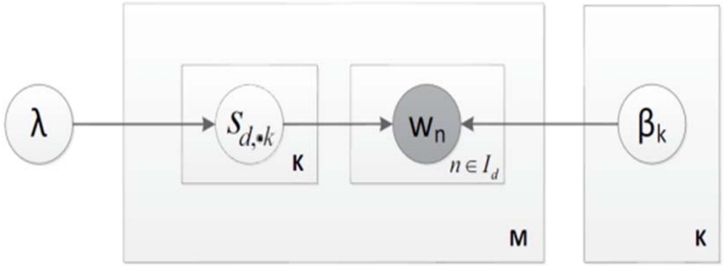
$$X_i | \sum_{j=1}^{n} X_j \sim Binom\left(\sum_{j=1}^{n} X_j, \frac{\tau_i}{\sum_{j=1}^{n} \tau_j}\right).$$

THEOREM 1. *Let θ be the topic proportion vector of document d. Assume the document is generated as described in section 3.1, we will have the kth topic proportion $\theta_k = \frac{\sum_{n=1}^{|I|} s_{nk}\beta_{kn}}{\sum_{n=1}^{|I|}\sum_{k=1}^{K} s_{nk}\beta_{kn}}$.*

$$\theta_k = \mathrm{E}\left[\frac{\sum_{n=1}^{|I|} w_{nk}}{\sum_{n=1}^{|I|}\sum_{k=1}^{K} w_{nk}} \Big| \sum_{n=1}^{|I|}\sum_{k=1}^{K} w_{nk}\right] = \frac{\mathrm{E}[\sum_{n=1}^{|I|} w_{nk} | \sum_{n=1}^{|I|}\sum_{k=1}^{K} w_{nk}]}{\sum_{n=1}^{|I|} w_n} = \frac{\sum_{n=1}^{|I|} s_{nk}\beta_{kn}}{\sum_{n=1}^{|I|}\sum_{k=1}^{K} s_{nk}\beta_{kn}}$$

$$\mathrm{E}[\sum_{n=1}^{|I|} w_{nk} | \sum_{n=1}^{|I|}\sum_{k=1}^{K} w_{nk}] = \left(\sum_{n=1}^{|I|} w_n\right)\left(\frac{\sum_{n=1}^{|I|} s_{nk}\beta_{kn}}{\sum_{n=1}^{|I|}\sum_{k=1}^{K} s_{nk}\beta_{kn}}\right)$$

Coding by GSTC



generated from M-Laplace distribution.

- Each topic produces some word occurrences from Poisson distribution.
- The occurrence of a word is the sum of occurrences from different topics, which follows the Poisson distribution too(see next slice)

Objective Function

$$argmin_{s,\beta} \sum_{d \in D} \Big(\sum_{n \in d} L(w_{d,n.}; s_{d,n.}^T \beta_{n.}) + \lambda \sum_{k=1}^{K} |s_{d,.k}|_2 + \lambda \sum_{n \in d} |s_{d,n.}|_1 \Big)$$

$$L(w_{d,n.}; s_{d,n.}^T \beta_{n.}) = \sum_{k=1}^{K} s_{d,nk} \beta_{nk} - w_{d,n.} \ln\Big(\sum_{k=1}^{K} s_{d,nk} \beta_{nk}\Big) + C$$

Algorithm

- Object function is bi-convex
  - Convex over s(word coding) or β(dictionary) when the other is fixed.
  - Learning s
    - Fixing β
    - Learning s using block-coordinate descent
  - Learning β
    - Fixing s
    - Learning β with projected quasi-newton
  - Iterating until converge

Code to Topics

$$\theta_k = \mathrm{E}\left[\frac{\sum_{n=1}^{|\mathbf{I}|} w_{nk}}{\sum_{n=1}^{|\mathbf{I}|}\sum_{k=1}^{K} w_{nk}} \,\middle|\, \sum_{n=1}^{|I|}\sum_{k=1}^{K} w_{nk}\right] = \frac{\mathrm{E}[\sum_{n=1}^{|I|} w_{nk} | \sum_{n=1}^{|I|}\sum_{k=1}^{K} w_{nk}]}{\sum_{n=1}^{|I|} w_n} = \frac{\sum_{n=1}^{|\mathbf{I}|} \mathbf{s}_{nk}\boldsymbol{\beta}_{kn}}{\sum_{n=1}^{|\mathbf{I}|}\sum_{k=1}^{K} \mathbf{s}_{nk}\boldsymbol{\beta}_{kn}}$$

$$\mathrm{E}[\sum_{n=1}^{|I|} w_{nk} | \sum_{n=1}^{|I|}\sum_{k=1}^{K} w_{nk}] = \left(\sum_{n=1}^{|I|} w_n\right)\left(\frac{\sum_{n=1}^{|I|} s_{nk}\beta_{kn}}{\sum_{n=1}^{|I|}\sum_{k=1}^{K} s_{nk}\beta_{kn}}\right)$$

- **Topic proportion can be re-constructed from the word codes and dictionary**

Poisson Distribution

Sums of Poisson

If $X_i \sim Poisson(\lambda_i),\, i = 1, \ldots, n$ are independent, and $\lambda = \sum_{i=1}^{n} \lambda_i$, then

$$Y = \sum_{i=1}^{n} X_i \sim Poisson(\lambda)$$

Moran’s Property

If $X_1, X_2, \ldots, X_n$ are independent Poisson random variables with parameters $\lambda_1, \lambda_2, \ldots, \lambda_n$, then

given $\sum_{j=1}^{n} X_i = k$, $X_i \sim Binom(k, \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j})$

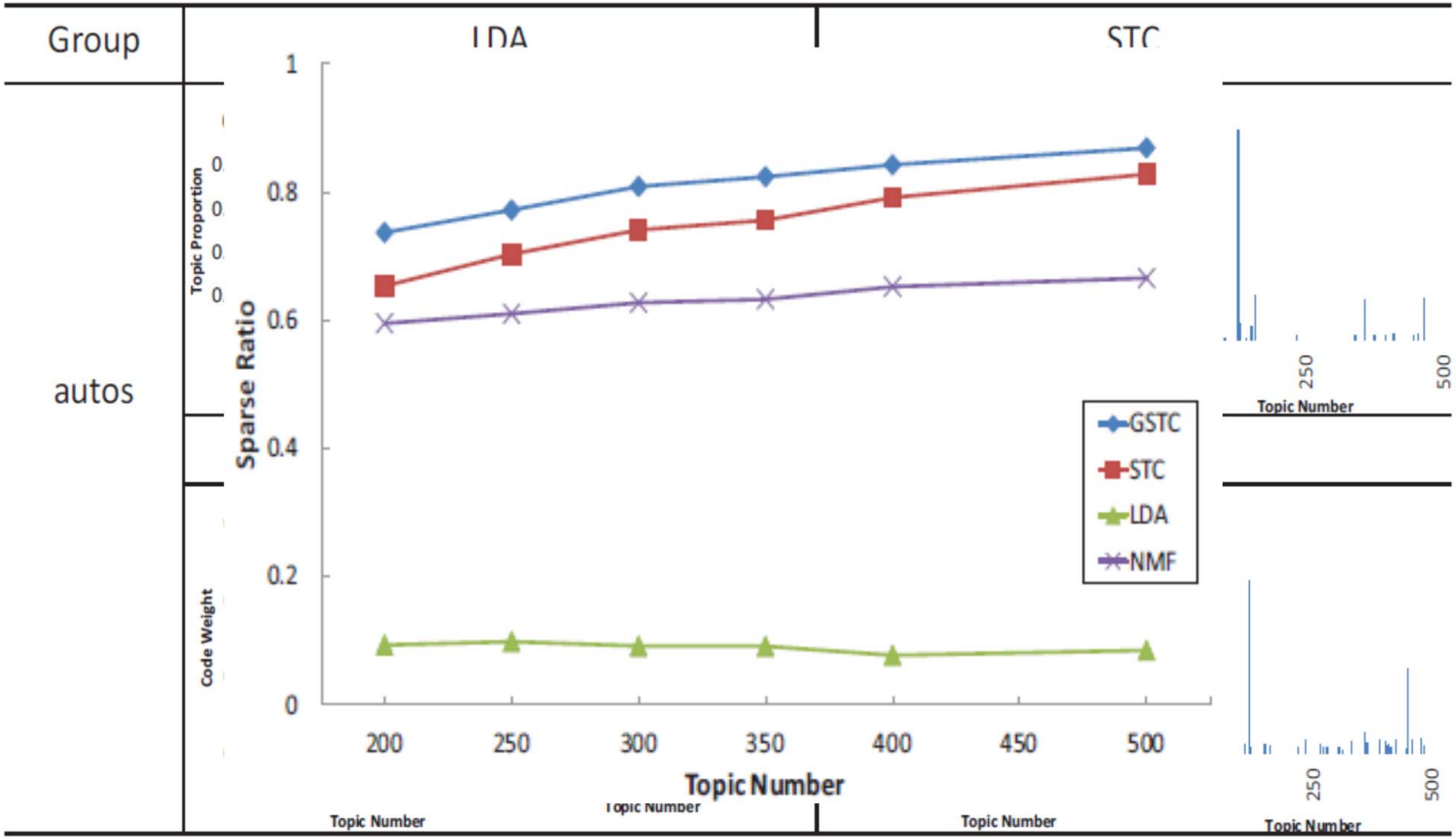
In fact,

$$\{X_i\} \sim Multinom(k, \left\{, \frac{\lambda_i}{\sum_{j=1}^{n} \lambda_j}\right\})$$

Results

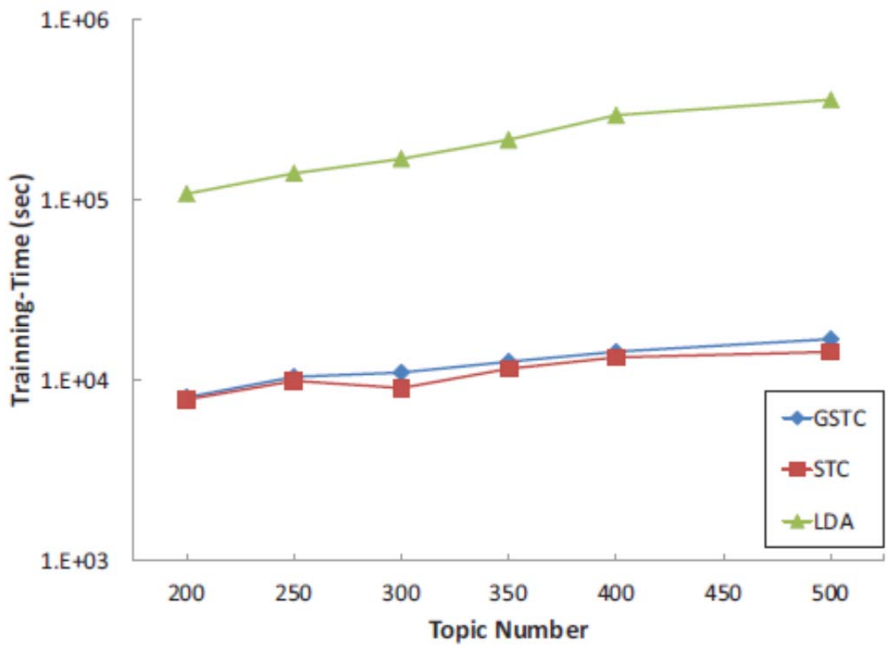
- Dataset
  - 20-newsgroup
    - 18,846 documents
    - 26,214 distinct words
    - 20 categories
- Baseline methods
  - LDA, NMF, STC
- Evaluation
  - Topic Sparsity
  - Training time
  - Classification accuracy

Topic Sparsity

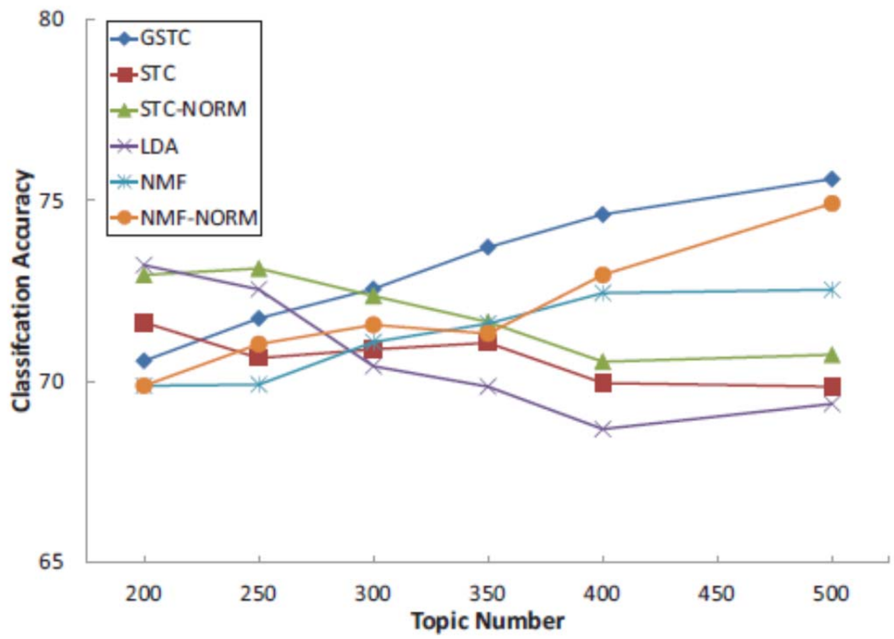


Effectiveness & Efficiency

Training Time

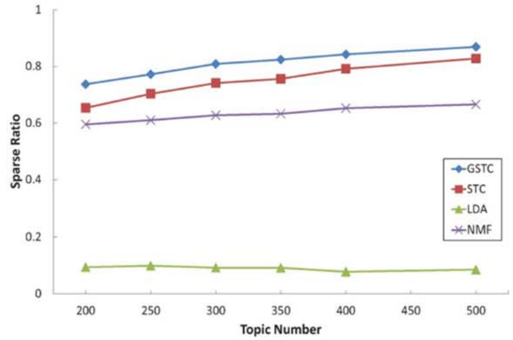


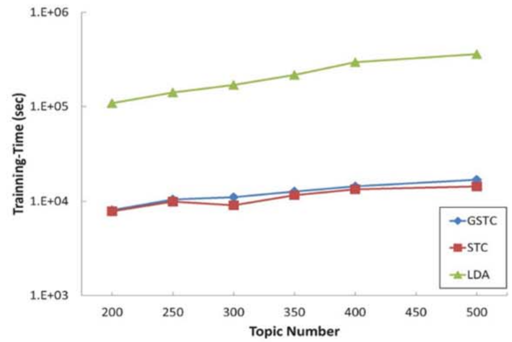
Classification Performance

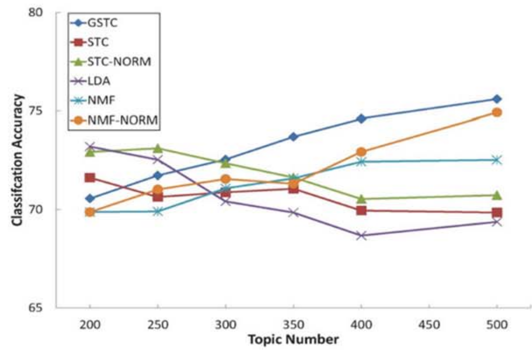


Results

- Dataset
  - 20-newsgoup
    - 18, 846 document,
    - 26, 214 distinct words
    - 20 related categories
- Baseline methods
  - LDA, NMF, STC
- Evaluation
  - Topic sparsity
  - Train time
  - Accuracy of document classification





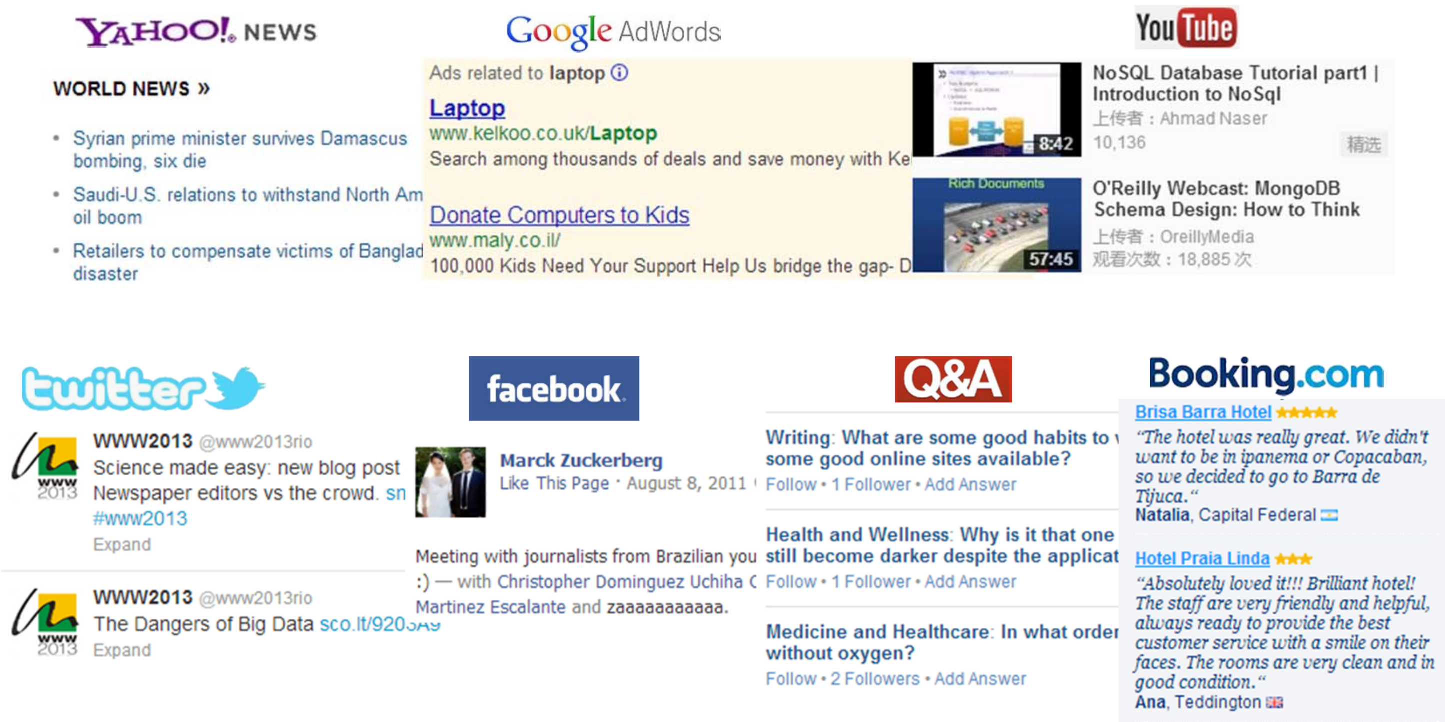


Topic Modeling (Feature Sparseness):

Biterm Topic model for Short Text (WWW'13,TKDE)

Short texts are prevalent

Uncovering the topics of short texts is crucial for a wide range of content analysis tasks



content characterizing **emerging topic detecting** **content recomendatio**

semantic analysis **user interest profiling** **…**

The limitation of conventional topic models

Bag-of-words Assumption

As for the Arabian and Palestinean voices that are against the current negotiations and the so-called peace process, they are not against peace per se, but rather for their well-founded predictions that Israel would NOT give an inch of the West bank (and most probably the same for Golan Heights) back to the Arabs. An 18 months of "negotiations" in Madrid, and Washington proved these predictions. Now many will jump on me saying why are you blaming israelis for no-result negotiations. I would say why would the Arabs stall the negotiations, what do they have to loose ?

- The occurrences of words play less discriminative role
  - Not enough word counts to know how words are related
- The limited contexts in short texts
  - More difficult to identify the senses of ambiguous words in short documents

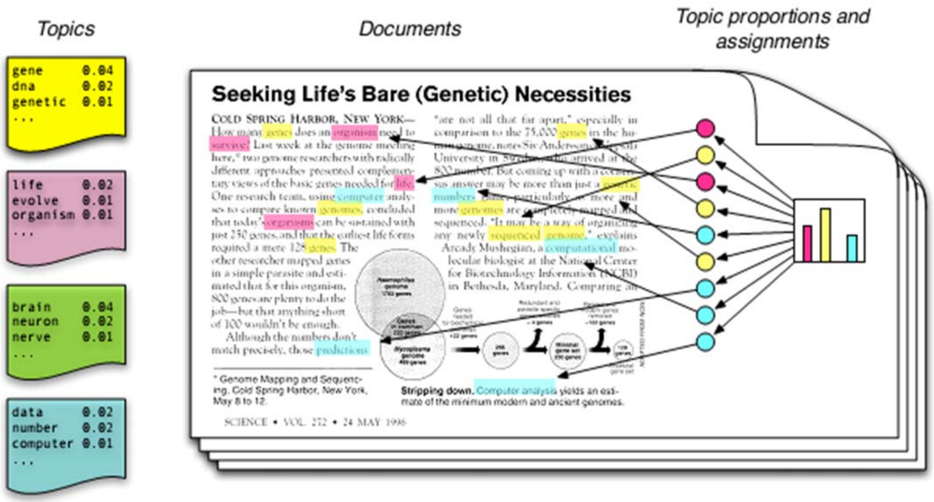
Previous Approaches

- LDA with document aggregation
  - e.g. aggregating the tweets published by the same user
  - heuristic, not general

- Mixture of unigrams
  - each document has only one topic
  - too strict assumption, result in peaked posteriors P(z|d)

- Sparse topic models
  - each dcoument maintains a sparse distribution over topics, e.g. Focused Topic Models
  - too complex, easy to overfitting

Key idea of our approach



- ✓ Since topics are basically groups of correlated words and the correlation is revealed by word co-occurrence patterns in documents, why not explicitly model the word co-occurrence for topic learning?

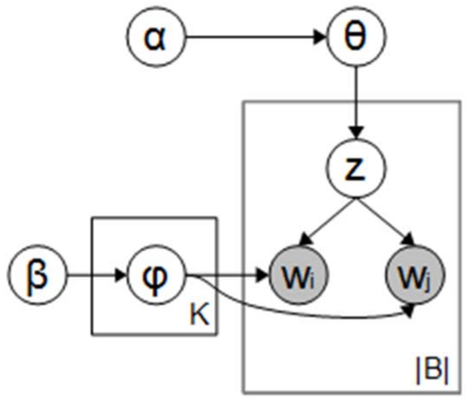
- ✓ Since topic models on short texts suffer from the problem of severe sparse patterns in short documents, why not use the rich global word co-occurrence patterns for better revealing topics?

Biterm Topic Model (BTM)

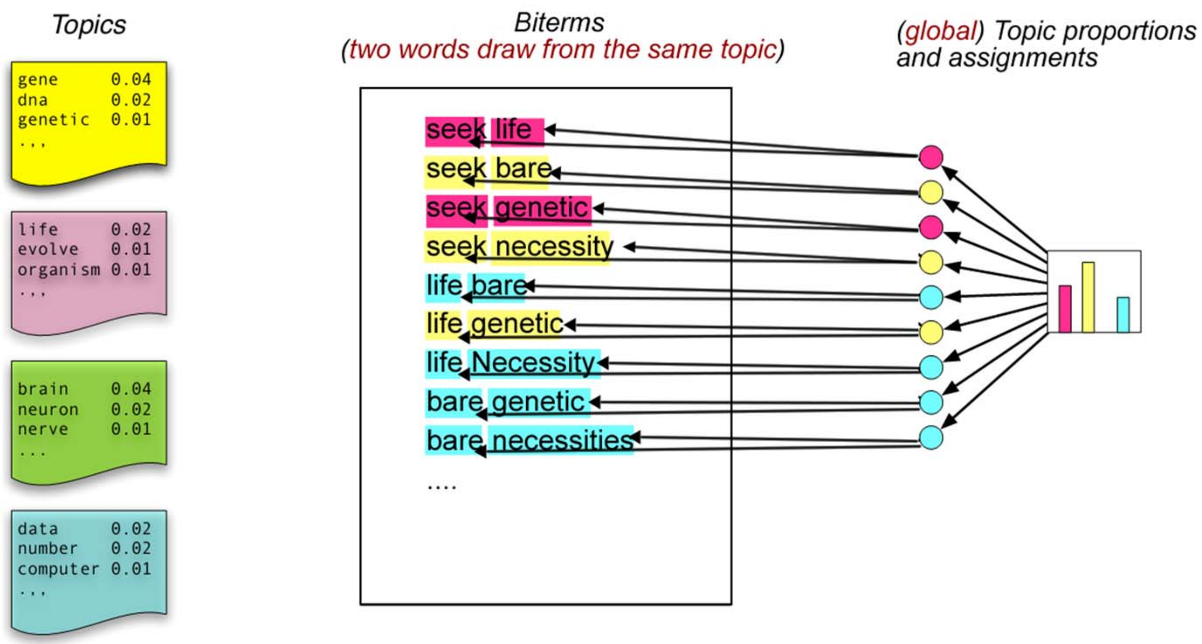
- BTM models the generation of word co-occurrences in a corpus
  - A biterm is an unordered word pair co-occurring in the same short context (document)
  - Training data includes all the biterms in the corpus

- Generative description

1. For each topic z
   (a) draw a topic-specific word distribution $\phi_z \sim Dir(\beta)$
2. Draw a topic proportion vector $\theta \sim Dir(\alpha)$ for the whole collection
3. For each biterm $\mathbf{b}$
   (a) draw a topic assignment $z \sim Multi(\theta)$
   (b) draw two words: $w_i, w_j \sim Mulit(\phi_z)$



Biterm Topic Model (BTM)



- Model the generation of biterms with latent topic structure
  - a topic ~ a probability distribution over words
  - a corpus ~ a mixture of topics
  - a biterm ~ two i.i.d sample drawn from one topic

Inferring Topics in a Document

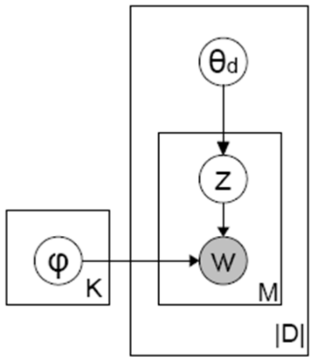
- Assumption
  - the topic proportions of a document equals to the expectation of the topic proportions of biterms in it

$$P(z|d) = \sum_b P(z|b)P(b|d)$$

where

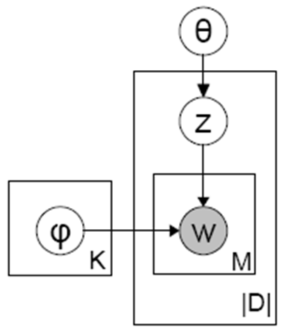
$$P(z|b) = \frac{P(z)P(w_i|z)P(w_j|z)}{\sum_z P(z)P(w_i|z)P(w_j|z)}, \qquad P(b|d) = \frac{n_d(b)}{\sum_b n_d(b)}$$

Comparison between different models



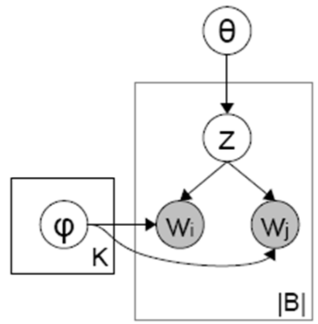
(a)

LDA



(b)

Mixture of Unigram



(c)

BTM

- Document level topic distribution
  - Suffer sparsity of the doc
- Model the generation of each word
  - Ignore context

- Corpus level topic distribution
  - Alleviate doc sparsity
- Single topic assumption in each document
  - Too strong assumption

- Corpus level topic distribution
  - Alleviate doc sparsity
- Model the generation of word pairs
  - Leverage context

Evaluation on Tweets

- Dataset：Tweets2011
  - Sample 50 hashtag with clear topic
  - Extract tweets with these hashtags

- Evaluation Metric: H score

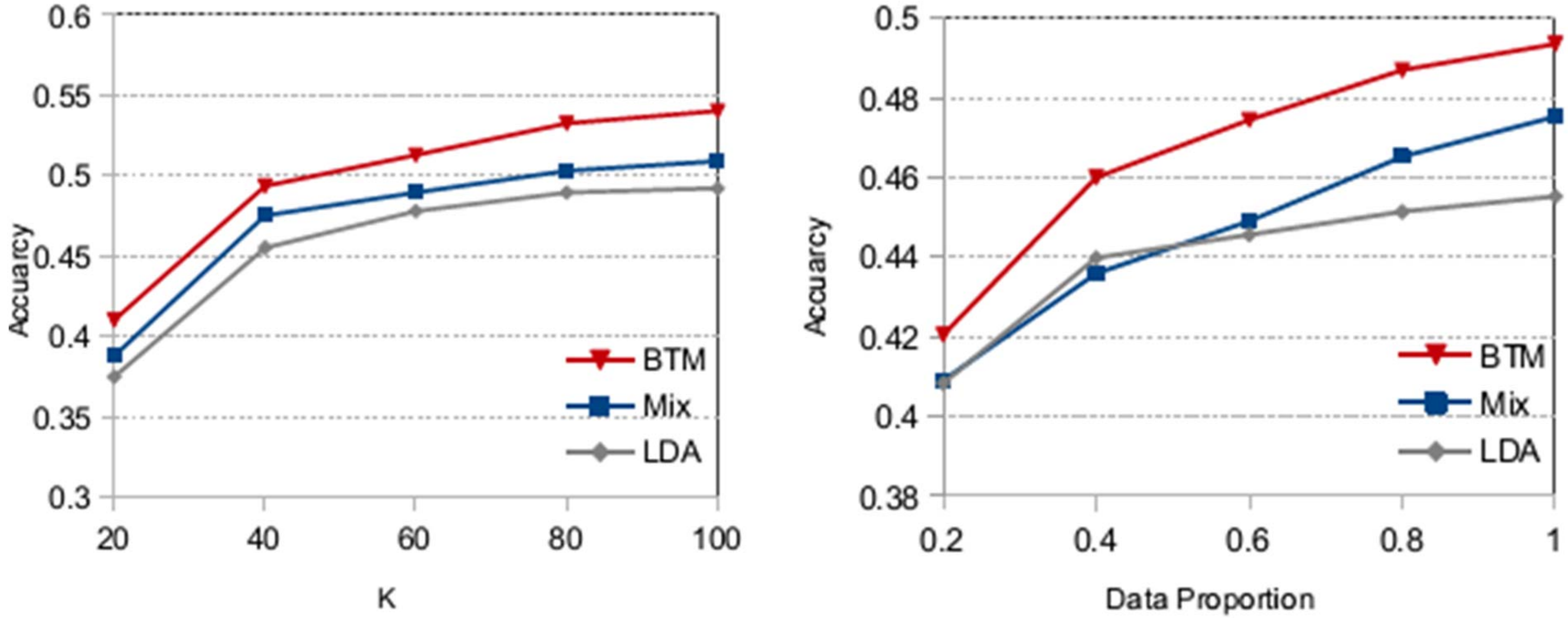
$$H = \frac{IntraDis(C)}{InterDis(C)}.$$

  - IntraDis: average distance between docs under the same hashtag
  - InterDis: average distance between docs under different hashtags
  - The smaller H score is, the better topic representation

Method	H score	Significant differences
LDA	0.576 ± 0.007	
LDA-U	0.564 ± 0.011	>LDA*
Mix	0.503 ± 0.008	>LDA-U**>LDA***
BTM	0.474 ± 0.005	>Mix***>LDA-U***>LDA***

Evaluation on Baidu Zhidao

- Dataset：Baidu Zhidao Q&A
  - Question classification according to their tags



Part III

Learning to Rank

Ranking is a Central Problem!

Web Search

Ranking

Information Filtering

Recommendation

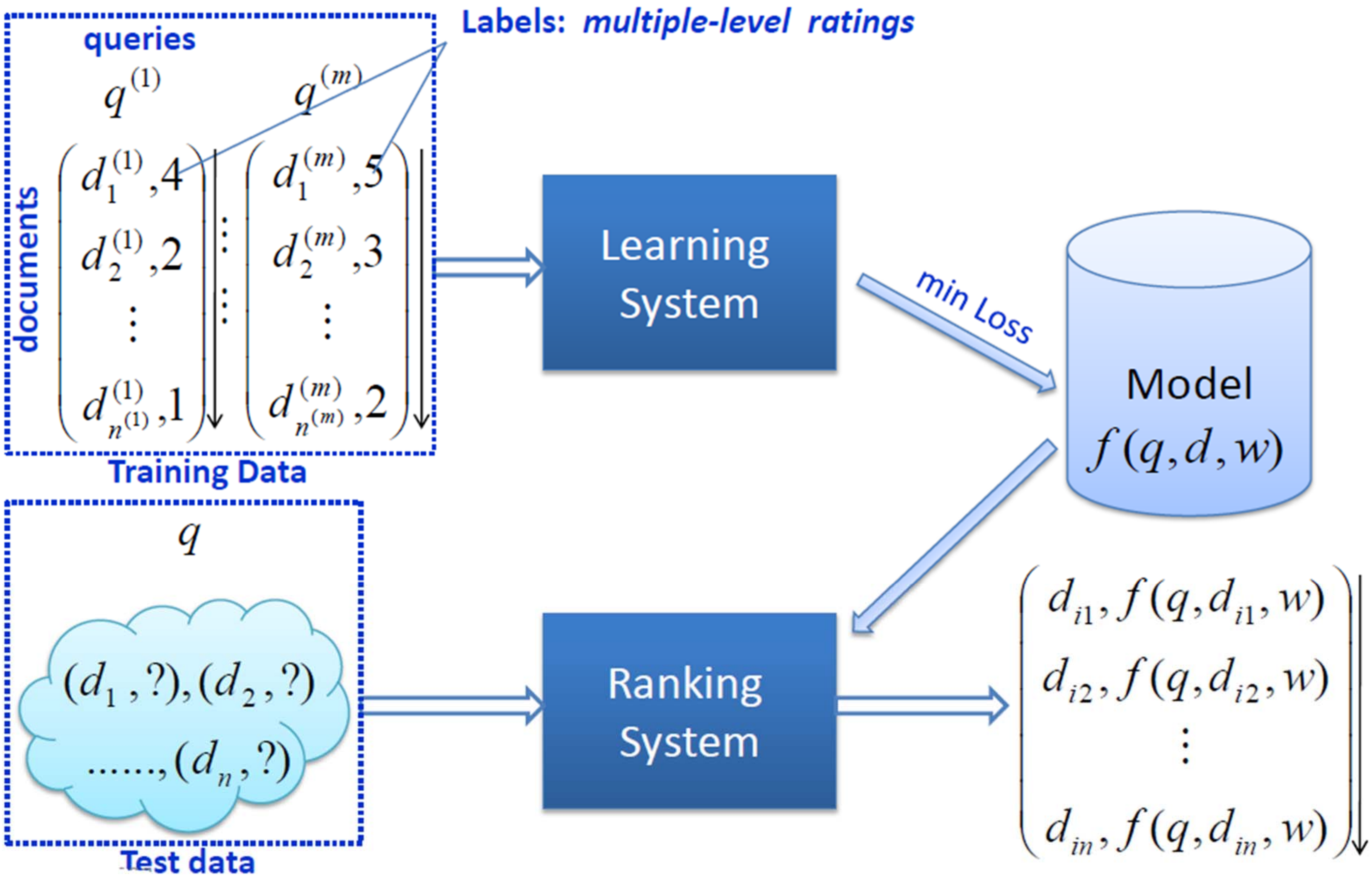
Conventional Methods

- Query-Relevant Methods
  - Boolean Algebra;
  - Latent Factor Indexing (LSI)
  - BM25, Language Model
- Query-Irrelevant Methods
  - Link Analysis (PageRank)

- **How to combine?**
  - **Parameter Tuning**
  - **Over-fitting**

Machine Learning Can Help!

Bring Machine Learning to Ranking



Learning to Rank Algorithms

$$x \rightarrow y$$

Pointwise Methods

- **Regression， Order Regression**
- **OC SVM, McRank**

$$(x_1, x_2) \rightarrow y$$

Pairwise Methods

- **Pairwise classification**
- **RankSVM， RankBoost， RankNet， GBRank**

$$(x_1, x_2, \cdots, x_n) \rightarrow \vec{y}$$

Listwise Methods

- **Listwise ranking**
- **ListMLE， ListNet， RankCosine， StructureSVM， SoftRank， AdaRank**

Evaluation Measures

- **Idea：Get the Right Ranking of High Relevant Documents**
  - **MAP:**

$$P@k = \frac{\#\{\text{relevant documents in top } k \text{ results}\}}{k}$$

$$AP = \frac{\sum_k P@k \cdot l_k}{\#\{\text{relevant documents}\}}$$

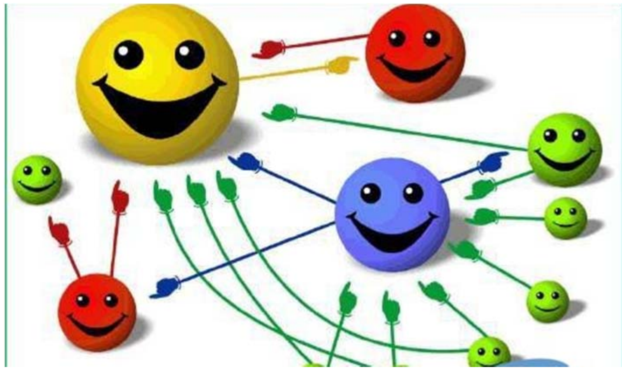
  - **NDCG:**

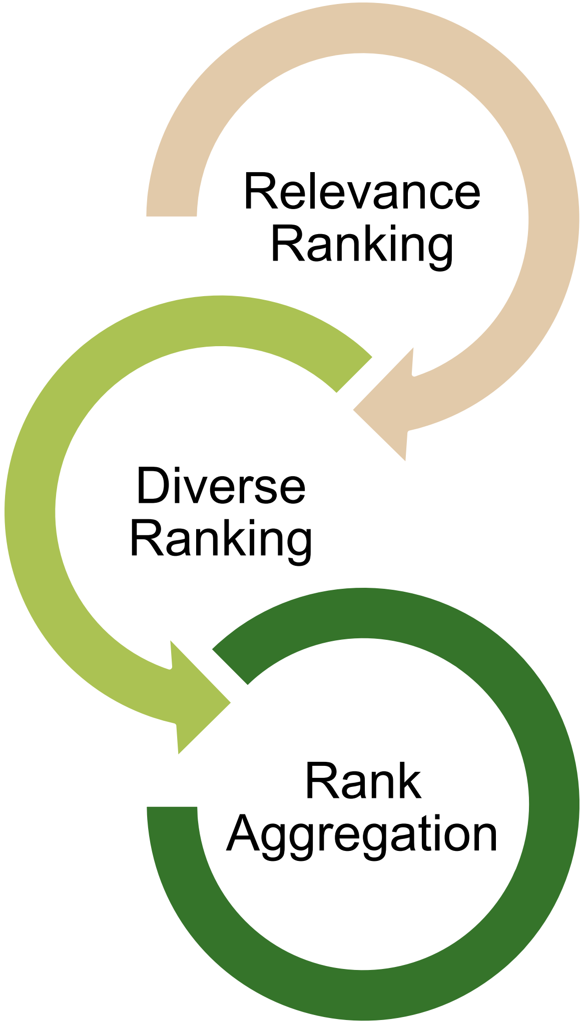
  - **ERR:**

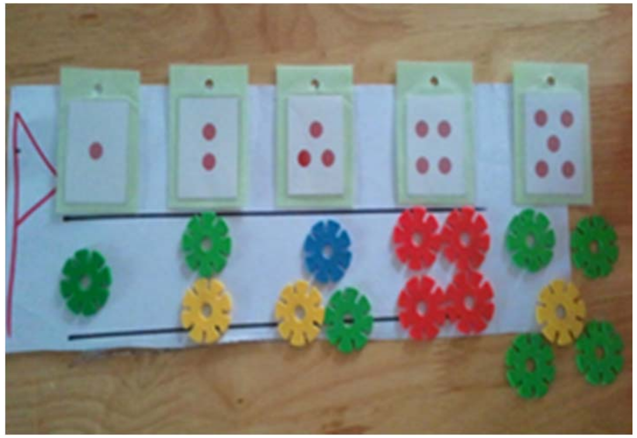
$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)} \qquad \mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p}$$

$$ERR = \sum_{i=1}^{n} \frac{1}{n} R(r_i) \prod_{j=1}^{i-1} (1 - R(r_j)),\ R(r) = \frac{2^r - 1}{16},$$

Outlines: Our Work







Relevance Ranking: Top-k Learning to Rank

Top-k Learning to Rank: Labeling, Ranking and Evaluation (SIGIR2012 Best Student Paper)
Statistical Consistency of Ranking Methods in a Rank-Differentiable Probability Space (NIPS2012)
A New Probabilistic Model for Top-k Ranking Problem (CIKM2013)
Is Top-k Sufficient for Ranking?(CIKM2013)
What Makes Data Noise: A Data Analysis in Learning to Rank (SIGIR2014)
Positional-Aware ListMLE: A Sequential Learning Process for Ranking (UAI2014)
What Noise Affects Algorithm Robustness for Learning to Rank (Information Retrieval Journal 2015)

Motivation

One great challenge for learning to rank: it is difficult to obtain reliable training data from human assessors!

Absolute Relevance Judgment $x \Rightarrow y$

Relevance Score

Drawbacks:

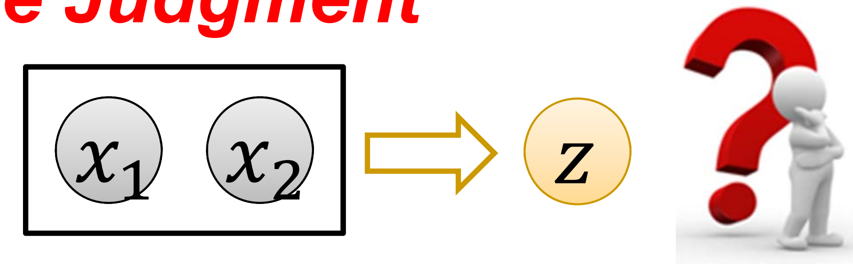
(1) Choice of the specific of the gradations.

(2) Increasing assessing burdens.

(3) High level of disagreement on judgments.

Motivation (cont')

Pairwise Preference Judgment



Preference Order

Pros:

(1) No need to determine the gradation specifications.
(2) Easier for an assessor to express a preference.
(3) Noise may be reduced.

Cons:

Complexity of judgment increases! (From O(n) to $O(n^2)$, O(n log n).)

How to reduce the complexity of pairwise preference judg

Motivation (cont')

- Do we really need to get a total ordering for each query? ***NO!***
- Users mainly care about the top results in real web search application!

Take more effort to figure out the top results and judge the preference orders among them.

Top-K Ground-truth

$$\begin{pmatrix} \boxed{\begin{matrix} x_1 \\ x_3 \\ x_5 \end{matrix}} \\ x_2, x_4 \end{pmatrix}$$

Total ordering of top K resu

Preferences between top
Documents and the other
N-K documents

Motivation (cont')

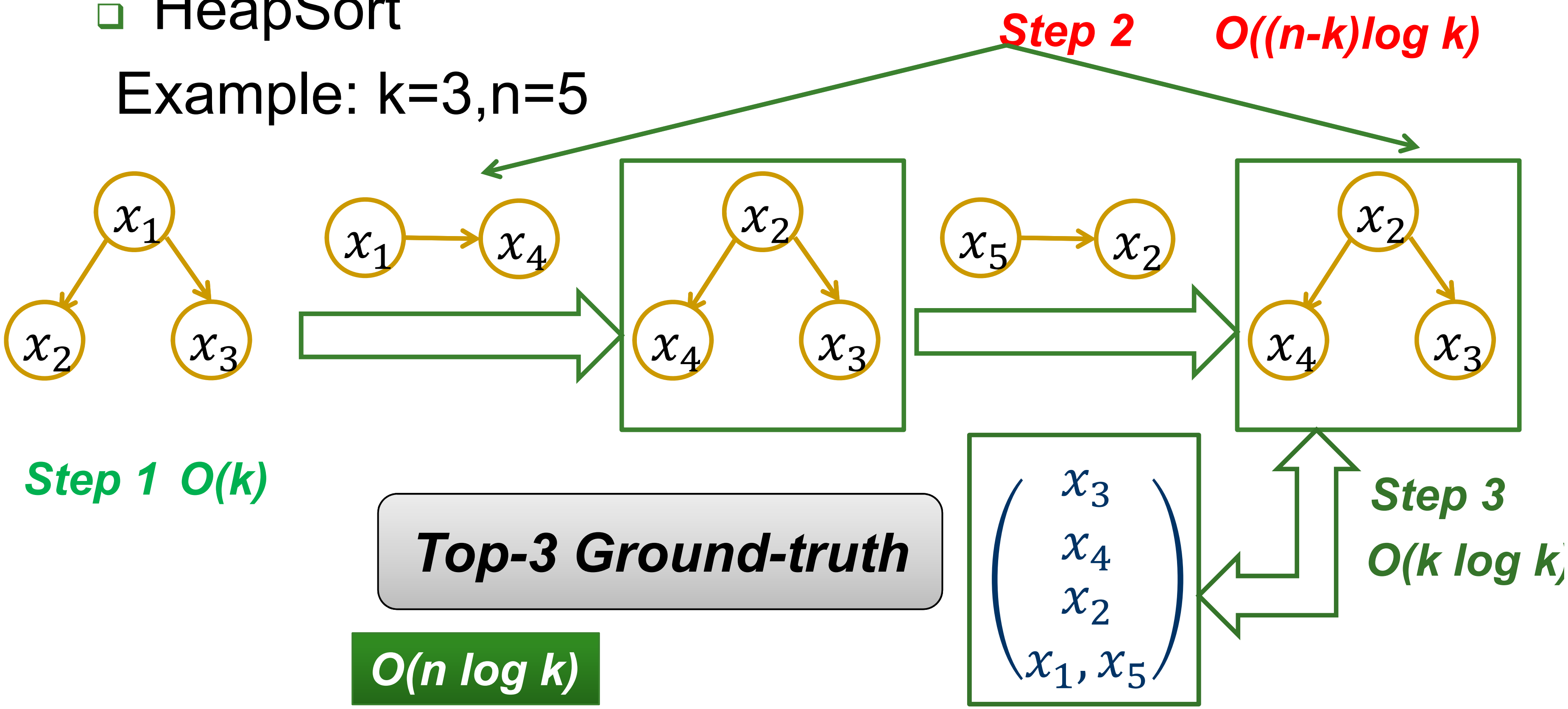
- Three Tasks:
  - How to design an efficient pairwise preference labeling strategy to get top-k ground-truth?
  - How to develop more powerful ranking algorithms in the new scenario?
  - How to define new evaluation measures for the new scenario?

Top-K Learning to Rank

Top-k Learning to Rank：Labeling

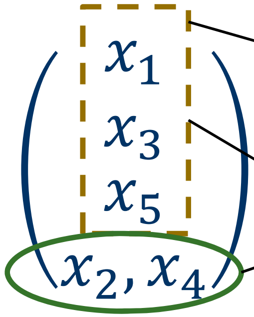
- Top-k Labeling Strategy
  - Pairwise preference judgment
  - HeapSort

Example: k=3,n=5



Top-K Learning to Rank：Ranking

- New characteristics of top-k ground-truth



Total ordering of top k items ⇨ *Listwise ranking algorithms*

Preferences between top k Items and the other n-k items ⇨ *Pairwise ranking algorithms*

$$L(f;q_i)=\beta\times L_{list}(f;T_i,\mathbf{y}_i)+(1-\beta)\times L_{pair}(f;P_i,\mathbf{y}_i)$$ ***FocusedRank***

Struct-SVM	**RankSVM**	**FocusedSVM**
AdaRank	**RankBoost**	**FocusedBoost**
ListNet	**RankNet**	**FocusedNet**

Top-K Learning to Rank: Evaluation

- Traditional evaluation measures, e.g. MAP, NDCG, ERR, are mainly defined on absolute relevance scores.
- In the scenario of top-k ground-truth, define a position-aware relevance score:

$$y_j^{(i)} = k + 1 - \pi_i(x_j^{(i)}), \text{ if } x_j^{(i)} \in T_i, \ \ y_j^{(i)} = 0, \text{ otherwise.}$$

  - *κ-NDCG* $$\kappa - NDCG@l = \frac{1}{N_l'} \sum_{j=1}^{l} \frac{2^{y_j^{(i)}} - 1}{\log_2(1+j)},$$
  - *κ-ERR* $$\kappa - ERR = \sum_{s=1}^{n} \frac{1}{n_i} R(y_s^{(i)}) \prod_{t=1}^{s-1} (1 - R(y_t^{(i)})), \ R(r) = \frac{2^r - 1}{2^{y_m^{(i)}}},$$

Experiments

- Effectiveness and efficiency of top-k labeling strategy
  - Data Sets: all the 50 queries from Topic Distillation task of TREC 2003, for each query, sample 50 documents.
  - Labeling Tools: top-10 labeling tool T1 and five-graded relevance judgment tool T2.
  - Assessors: Five graduate students who are familiar with web search.
  - Assignment: Divided into five folds Q1,…Q5, Ui judges Qi with T1 and Qi+1 with T2, for i=1,2,3,4, and U5 judges Q5 with T1 and Q1 with T2.

Experimental Results I

- Time Efficiency

Table 1: Comparison results of time efficiency

Method	Time per judgment(s)	Time per query(min)	Judgment complexity	#Judgments per query
Top-k labeling	5.51	13.13	$\mathcal{O}(n \log k)$	142.76
Five-grade judgment	13.87	11.78	$\mathcal{O}(n)$	50

- Agreement

	A$\succ$B	A$\sim$B	A$\prec$B
A$\succ$B	0.6749	0.2766	0.0485
A$\sim$B	0.1138	0.8198	0.0664
A$\prec$B	0.1047	0.3779	0.5174

Top 10 Labeling

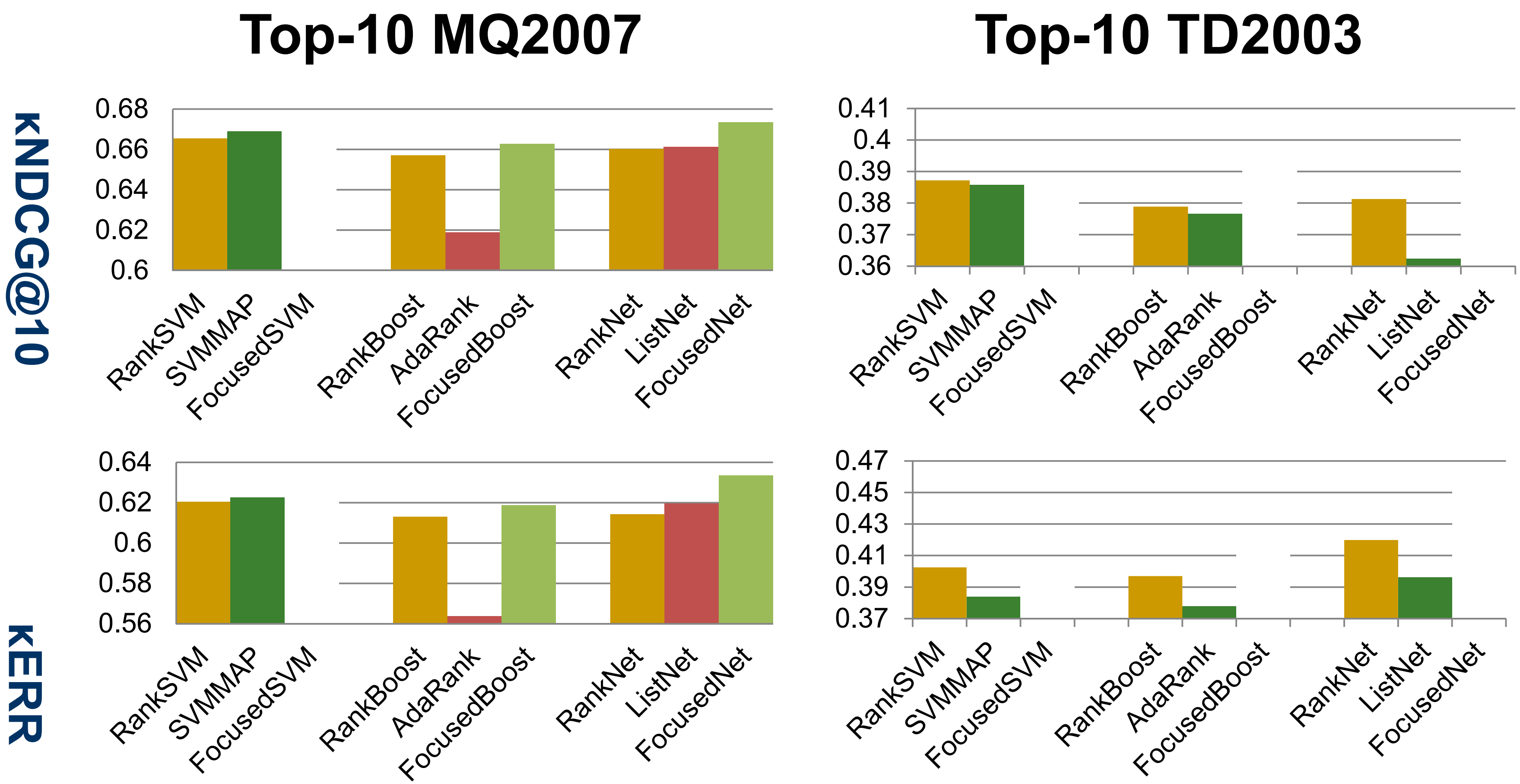
	A$\succ$B	A$\sim$B	A$\prec$B
A$\succ$B	0.6272	0.2913	0.0815
A$\sim$B	0.2825	0.5232	0.1944
A$\prec$B	0.1534	0.3826	0.4640

5 Graded Labeling

Experiments (cont')

- Performance of FocusedRank
  - Baselines:
    (1) Pairwise: RankSVM, RankBoost, RankNet,
    (2) Listwise: SVMMAP, AdaRank, ListNet,
    (3) Top-k: Top-k ListMLE
  - Data Sets:
    (1) MQ2007 (From LETOR): Graded MQ2007 and Top-k MQ2007
    (2) TD2003 (Previous constructed data): Graded TD2003 and Top-k TD2003

Experimental Results II

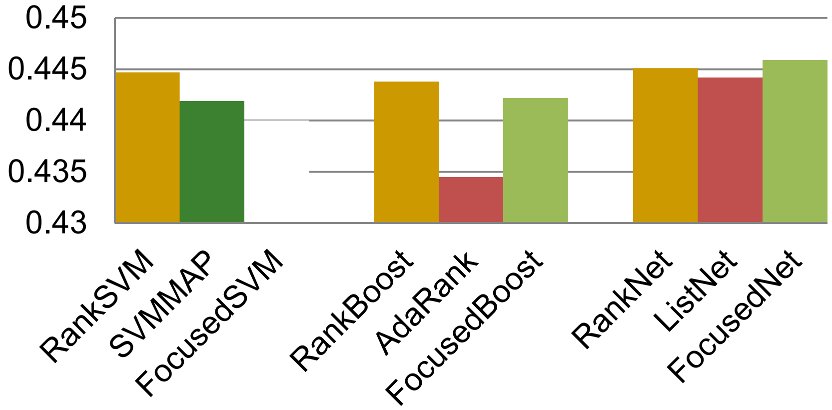


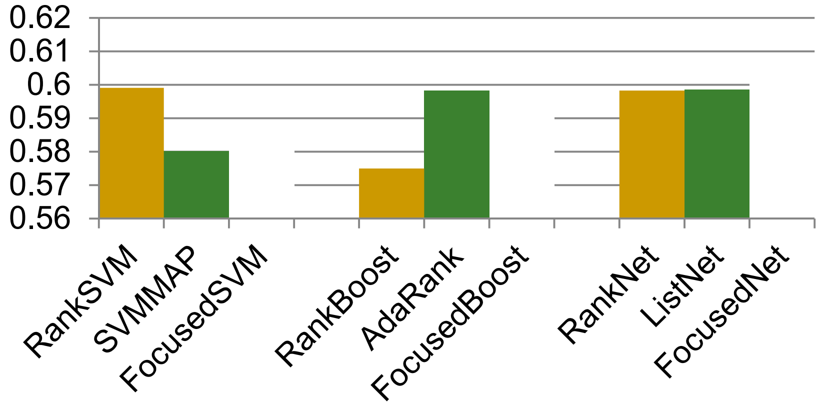
Performance comparison among ocusedRank, pairwise and listwise algorithms on Top-k datase

Experimental Results II (cont')

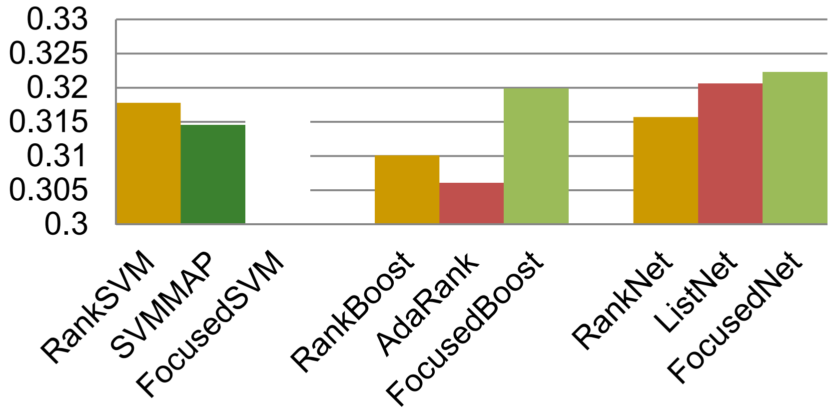
Graded MQ2007 | **Graded TD2003**

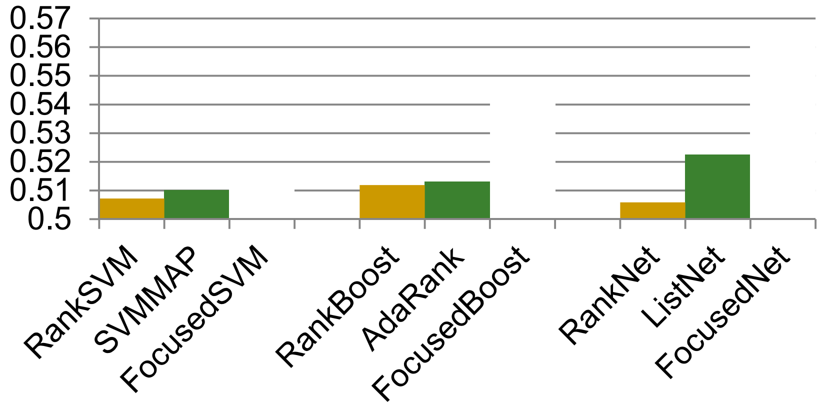
NDCG@10





ERR

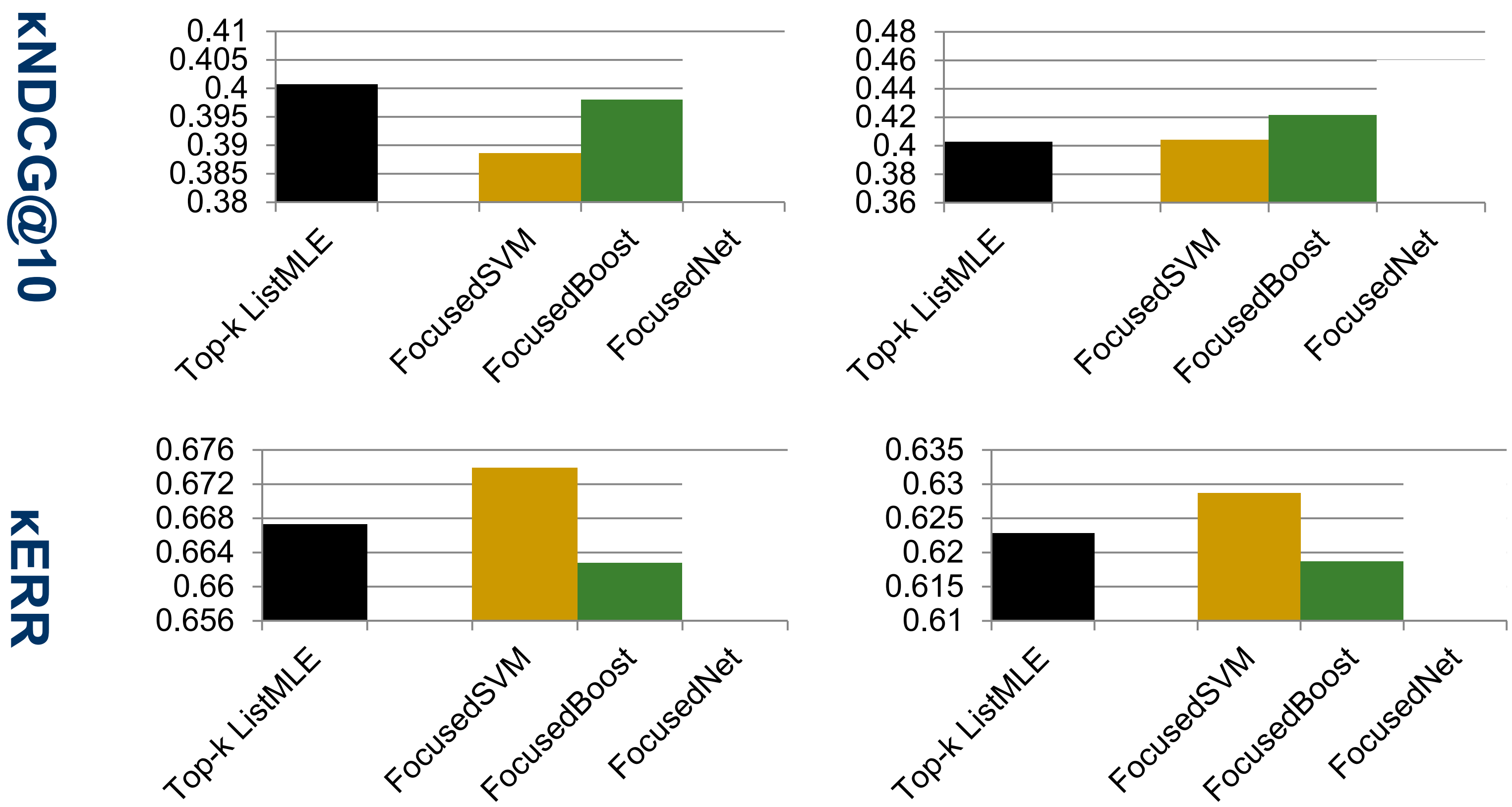




Performance comparison among FocusedRank, pairwise and listwise algorithms on Graded datasets.

Experimental Results II (cont')

Top-10 MQ2007 | **Top-10 TD2003**



Performance comparison between FocusedRank and Top-k ListMLE on Top-k datasets.

Is Top-k Sufficient for Ranking?

- Revisit the training of learning to rank:

Ideal — **Full-Order Ranking Lists**

$$\begin{pmatrix} x_{i_1} \\ x_{i_2} \\ \vdots \\ x_{i_{n-1}} \\ x_{i_n} \end{pmatrix}$$

User mainly care about top results !

$\approx$

Surrogate — **Top-k Ground-truth**

$$\begin{pmatrix} x_{i_1} \\ x_{i_2} \\ \vdots \\ x_{i_{k-1}} \\ x_{i_k} \\ x_{i_{k+1}}, \ldots x_{i_n} \end{pmatrix}$$

Assumption: top-k ground-truth is sufficient for ranking!

- Top-k labeling strategy based on pairwise preference judgment:

$\boxed{x_1 \quad x_2} \Rightarrow z$

Preference Order

HeapSort

- **The training data are proven to be more reliable! [SIGIR2012,CIKM2012]**

Best Student Paper Award

Empirical Study

Assumption: top-k ground-truth is sufficient for ranking!

⇕

Training on top-k setting is as good as that in full-order setting.

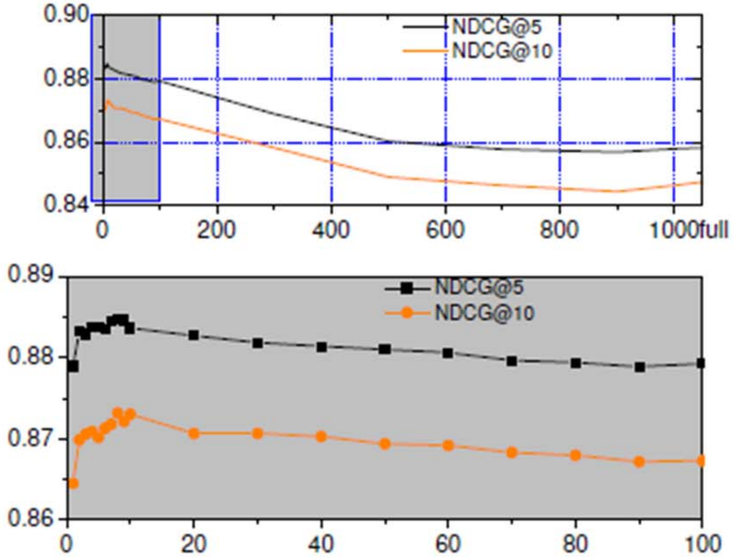
Ranking function f1

Ranking function f2

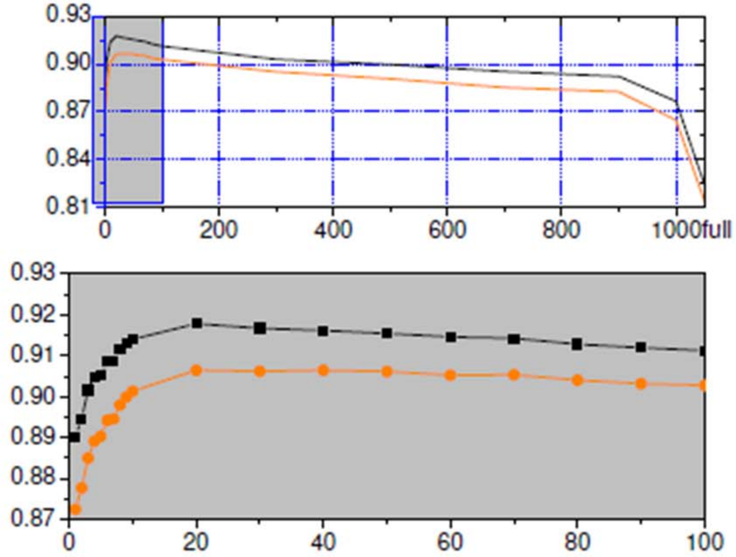
Test Performance Comparison

Experimental Setting

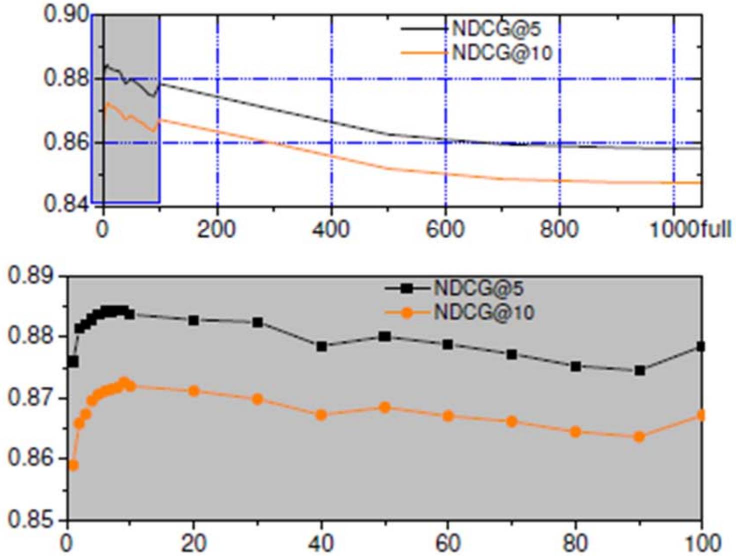
- Datasets
  - LETOR 4.0(MQ2007-list, MQ2008-list)
    - Ground-truth: full order
    - Top-k ground-truth are constructed by just preserving the total order of top k items
- Algorithms
  - Pairwise: Ranking SVM, RankBoost, RankNet
  - Listwise: ListMLE
- Experiments
  - Study how the test performances of ranking algorithms change w.r.t. k in the training data of top-k setting.



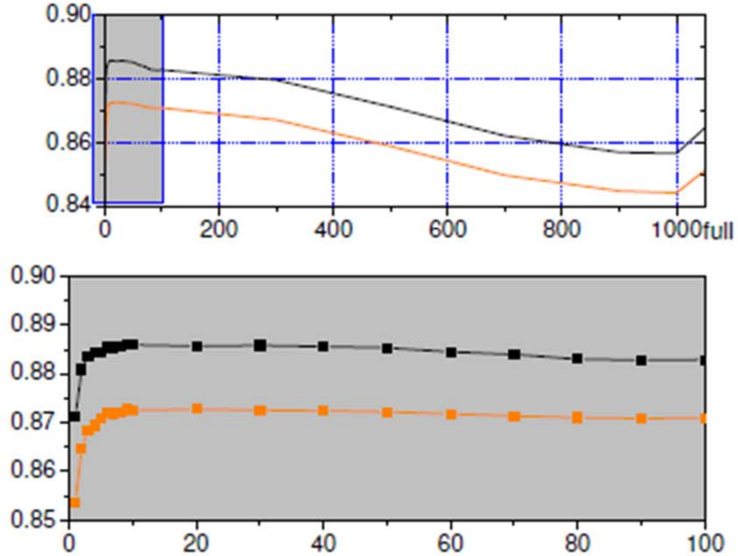
(a) Ranking SVM



(b) RankBoost

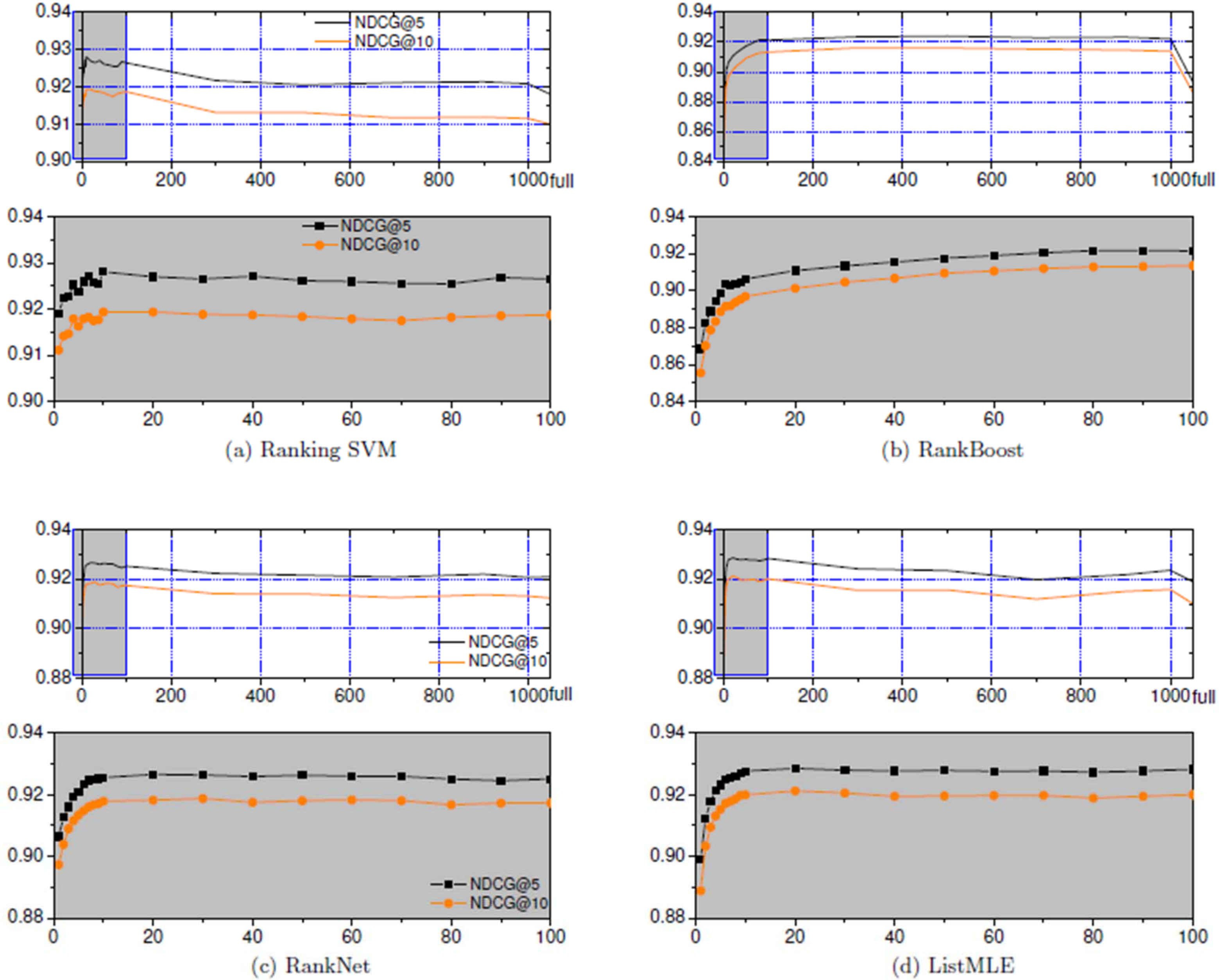


(c) RankNet



(d) ListMLE

Figure 1: Performance variations of different ranking algorithms in top-k setting on MQ2007-list with the increase of k



(a) Ranking SVM

(b) RankBoost

(c) RankNet

(d) ListMLE

Figure 2: Performance variation of different ranking algorithms in top-k setting on MQ2008-list with the increase of k

Experimental Results

(1) Overall, the test performance of ranking algorithms in top-k setting increase to a stable value with the growth of k.

(2) However, when k keeps increasing, the performances will decrease.

(3) The test performances of the four algorithms increase quickly to a stable value with the increase of k.

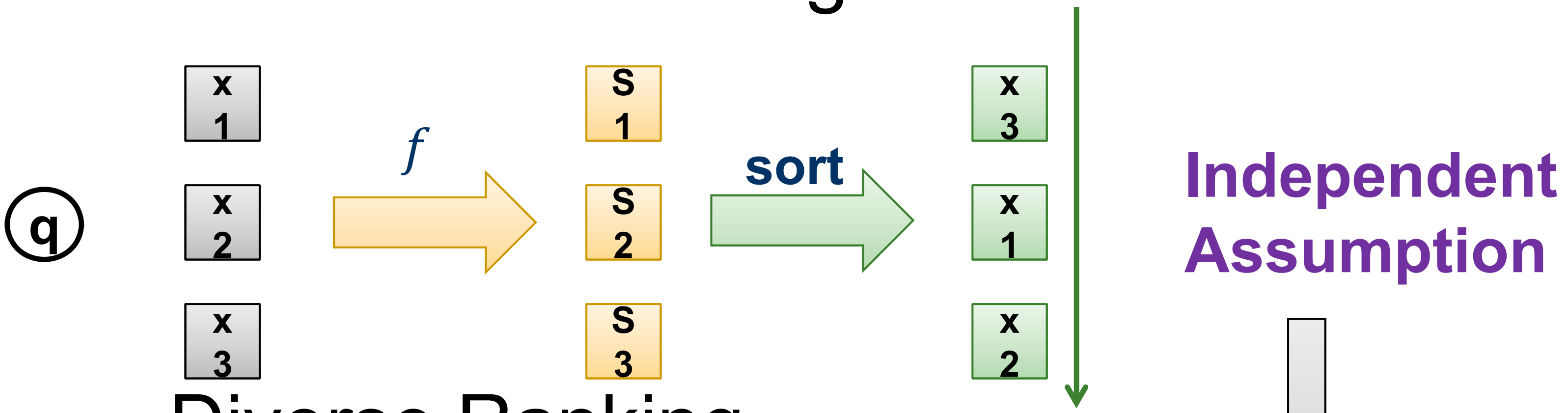
- Empirically, top-k ground-truth is sufficient for ranking!

Diverse Ranking: Relational Learning to Rank

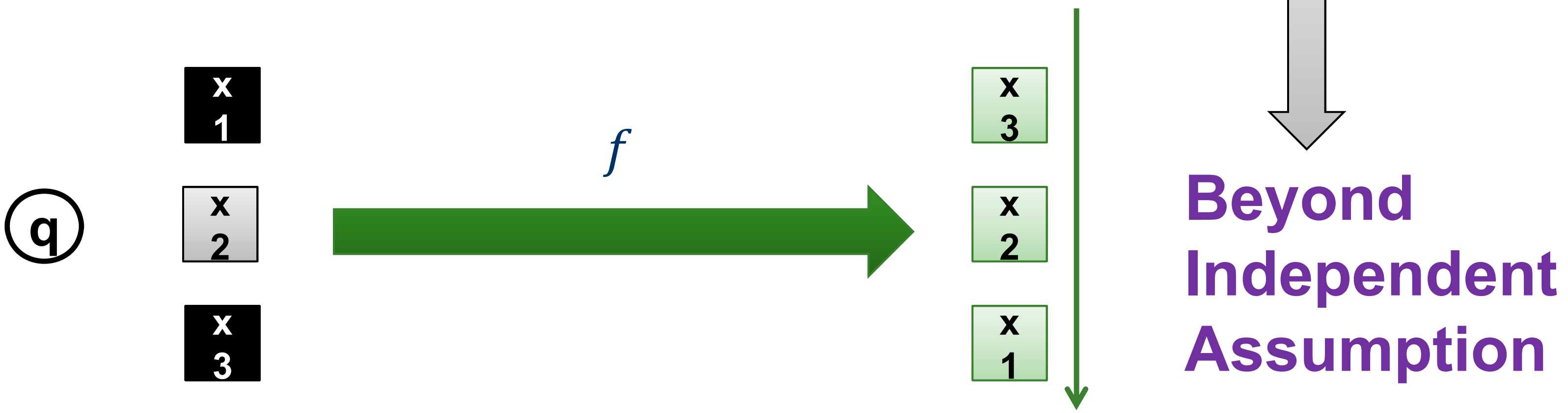
A Novel Relational Learning to Rank Approach for Topic-Focused Multi-Document Summarization (ICDM2013)
Learning for Search Result Diversification (SIGIR2014)
Learning Maximal Marginal Relevance Model via Directly Optimizing Diversity Evaluation Measures (SIGIR2015)

Beyond Relevance Ranking

- Relevance Ranking



- Diverse Ranking



Motivation

- Maximal Marginal Relevance

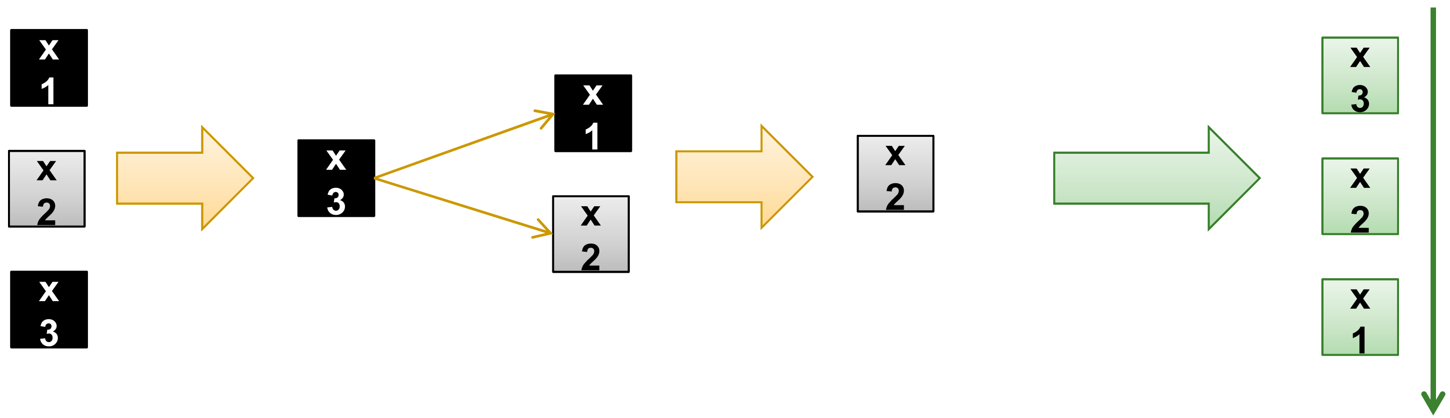
Non-Learning!

$$\mathrm{MMR} \stackrel{\mathrm{def}}{=} \mathop{\mathrm{Arg\,max}}_{D_i \in R \setminus S} \left[\lambda\, \mathrm{Sim}_1(D_i, Q) - (1-\lambda) \max_{D_j \in S} Sim_2(D_i, D_j) \right]$$

Relevance

Relation

  - Sequential Selection Procedure



Relational Learning to Rank

- Considering both content of individual objects and relations among objects.
- Formalization
  - Four key components: input space, out space, ranking function *f*, loss function *L*

Difference

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}\in\mathcal{F}} \sum_{i=1}^{N} L(\mathbf{f}(X^{(i)}, R^{(i)}), \mathbf{y}^{(i)}).$$

Definition of Ranking function

- Definition

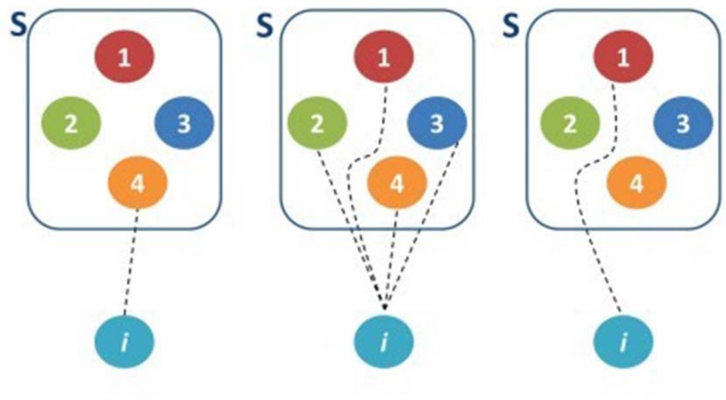
$$f_S(x_i, R_i) = \boxed{\omega_r^T \mathbf{x}_i} + \boxed{\omega_d^T h_S(R_i)}, \forall x_i \in X \setminus S$$

Content-based score

Relation-based score

- Relational function

$$h_S(R_i)$$



(a) Minimal (b) Average (c) Maximal

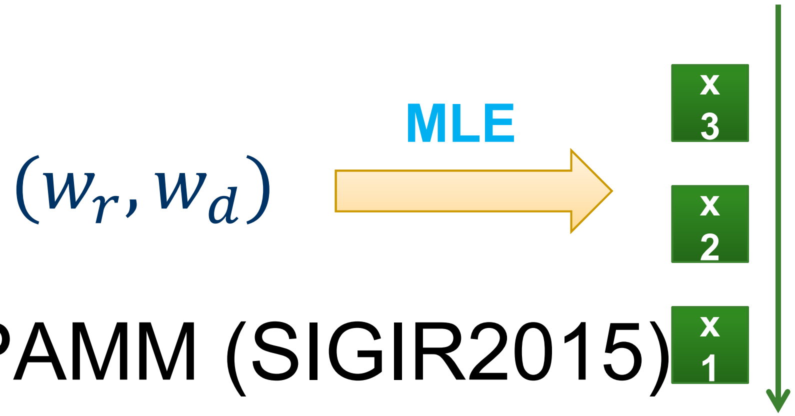
$$h_S(R_i) = \min_{x_j \in S} R_{ij}$$

$$h_S(R_i) = \frac{1}{|S|} \sum_{x_j \in S} R_{ij}$$

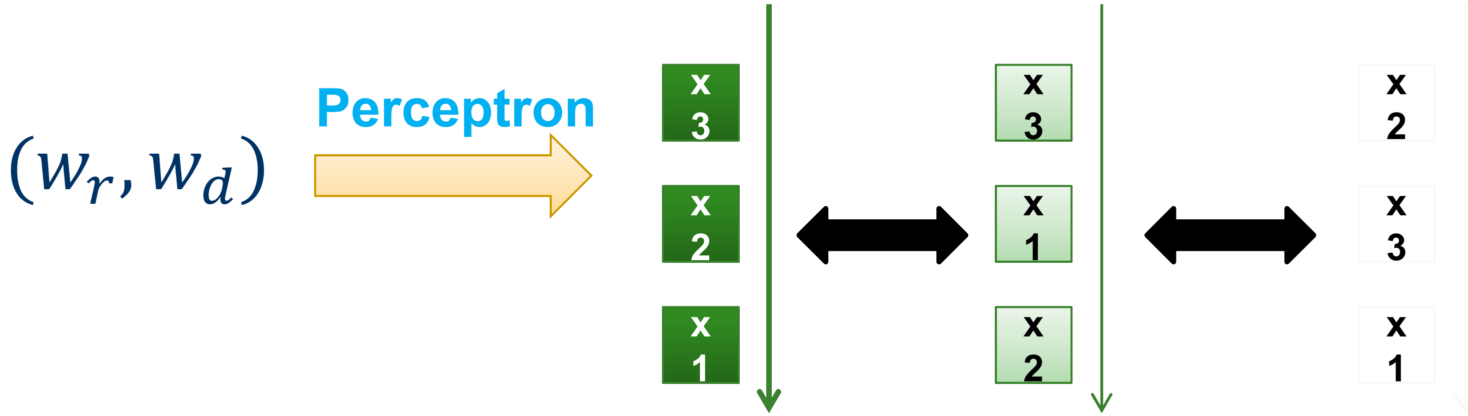
$$h_S(R_i) = \max_{x_j \in S} R_{ij}$$

Definition of Loss Function

- R-LTR (ICDM2013, SIGIR2014)



- PAMM (SIGIR2015)



Definition of Loss Function(R-LTR)

- **Plackett-Luce Model**

$$\mathbf{P}(\pi \mid \boldsymbol{v}) = \prod_{i=1}^{M} \frac{v_{\pi(i)}}{v_{\pi(i)} + v_{\pi(i+1)} + \ldots + v_{\pi(M)}}$$

- **Detailed definition**

$$P(x_{y(1)} \mid X) = \frac{\exp\{f_\phi(x_{y(1)})\}}{\sum_{k=1}^{n} \exp\{f_\phi(x_{y(k)})\}}, \quad P(x_{y(j)} \mid X \setminus S_{j-1}) = \frac{\exp\{f_{S_{j-1}}(x_{y(j)}, R_{y(j)})\}}{\sum_{k=j}^{n} \exp\{f_{S_{k-1}}(x_{y(k)}, R_{y(k)})\}}.$$

- **maximize the sum of the likelihood function**

$$-\sum_{i=1}^{N} \sum_{j=1}^{n_i} \log \left\{ \frac{\exp\{\omega_r^T \mathbf{x}_{y(j)}^{(i)} + \omega_d^T h_{S_{j-1}^{(i)}}(R_{y(j)}^{(i)})\}}{\sum_{k=j}^{n_i} \exp\{\omega_r^T \mathbf{x}_{y(k)}^{(i)} + \omega_d^T h_{S_{k-1}^{(i)}}(R_{y(k)}^{(i)})\}} \right\}$$

Definition of Loss Function(PAMM)

- Firstly, PAMM generates positive and negative rankings.
- Secondly, PAMM optimizes the model parameters ω_r and ω_d.

  1: $\Delta F \leftarrow F\left(X^{(n)}, R^{(n)}, \mathbf{y}^{+}\right) - F\left(X^{(n)}, R^{(n)}, \mathbf{y}^{-}\right)$

  2: **if** $\Delta F \leq E\left(X^{(n)}, \mathbf{y}^{+}, J^{(n)}\right) - E\left(X^{(n)}, \mathbf{y}^{-}, J^{(n)}\right)$

  3: **then**

  4: calculate $\nabla\omega_r^{(n)}$ and $\nabla\omega_d^{(n)}$

  5: $(\omega_r, \omega_d) \leftarrow (\omega_r, \omega_d) + \eta \times \left(\nabla\omega_r^{(n)}, \nabla\omega_d^{(n)}\right)$

  6: **end if**
- Finally, PAMM outputs the optimized model parameters (ω_r, ω_d).

Experiments

- Dataset: TREC WT2009, WT2010 and WT2011
- Data Processing
  - Indri toolkit (version 5.2)
  - Porter stemmer and stopwords removing
- Evaluation
  - TREC Official Measures: ERR-IA, a-NDCG
- Baselines:
  - QL, MMR, xQuAD, PM-2, ListMLE, SVMDIV

Feature Vectors

- Content-based features
  - Weighing features: VSM, BM25, LM..
  - Term dependency features: MRF
  - Length
  - Pos
  - …
- Relation-based features
  - Cosine diversity
  - Jaccard diversity
  - subtopic diversity
  - document-level co-occurrence
  - …

Experimental Results

Table 5: Performance comparison of all methods in official TREC diversity measures for WT2009.

Method	ERR-IA@20	α-NDCG@20
QL	0.164	0.269
ListMLE	0.191(+16.46%)	0.307(+14.13%)
MMR	0.202(+23.17%)	0.308(+14.50%)
xQuAD	0.232(+41.46%)	0.344(+27.88%)
PM-2	0.229(+39.63%)	0.337(+25.28%)
SVM-DIV	0.241(+46.95%)	0.353(+31.23%)
StructSVM(α-NDCG)	0.260(+58.54%)	0.377(+40.15%)
StructSVM(ERR-IA)	0.261(+59.15%)	0.373(+38.66%)
R-LTR	0.271(+65.24%)	0.396(+47.21%)
PAMM(α-NDCG)	0.284(+73.17%)	**0.427**(+58.74%)
PAMM(ERR-IA)	**0.294**(+79.26%)	0.422(+56.88%)

Table 6: Performance comparison of all methods in official TREC diversity measures for WT2010.

Method	ERR-IA@20	α-NDCG@20
QL	0.198	0.302
ListMLE	0.244(+23.23%)	0.376(+24.50%)
MMR	0.274(+38.38%)	0.404(+33.77%)
xQuAD	0.328(+65.66%)	0.445(+47.35%)
PM-2	0.330(+66.67%)	0.448(+48.34%)
SVM-DIV	0.333(+68.18%)	0.459(+51.99%)
StructSVM(α-NDCG)	0.352(+77.78%)	0.476(+57.62%)
StructSVM(ERR-IA)	0.355(+79.29%)	0.472(+56.29%)
R-LTR	0.365(+84.34%)	0.492(+62.91%)
PAMM(α-NDCG)	0.380(+91.92%)	**0.524**(+73.51%)
PAMM(ERR-IA)	**0.387**(+95.45%)	0.511(+69.21%)

Table 7: Performance comparison of all methods in official TREC diversity measures for WT2011.

Method	ERR-IA@20	α-NDCG@20
QL	0.352	0.453
ListMLE	0.417(+18.47%)	0.517(+14.13%)
MMR	0.428(+21.59%)	0.530(+17.00%)
xQuAD	0.475(+34.94%)	0.565(+24.72%)
PM-2	0.487(+38.35%)	0.579(+27.81%)
SVM-DIV	0.490(+39.20%)	0.591(+30.46%)
StructSVM(α-NDCG)	0.512(+45.45%)	0.617(+36.20%)
StructSVM(ERR-IA)	0.513(+45.74%)	0.613(+35.32%)
R-LTR	0.539(+53.13%)	0.630(+39.07%)
PAMM(α-NDCG)	0.541(+53.70%)	**0.643**(+41.94%)
PAMM(ERR-IA)	**0.548**(+55.68%)	0.637(+40.62%)

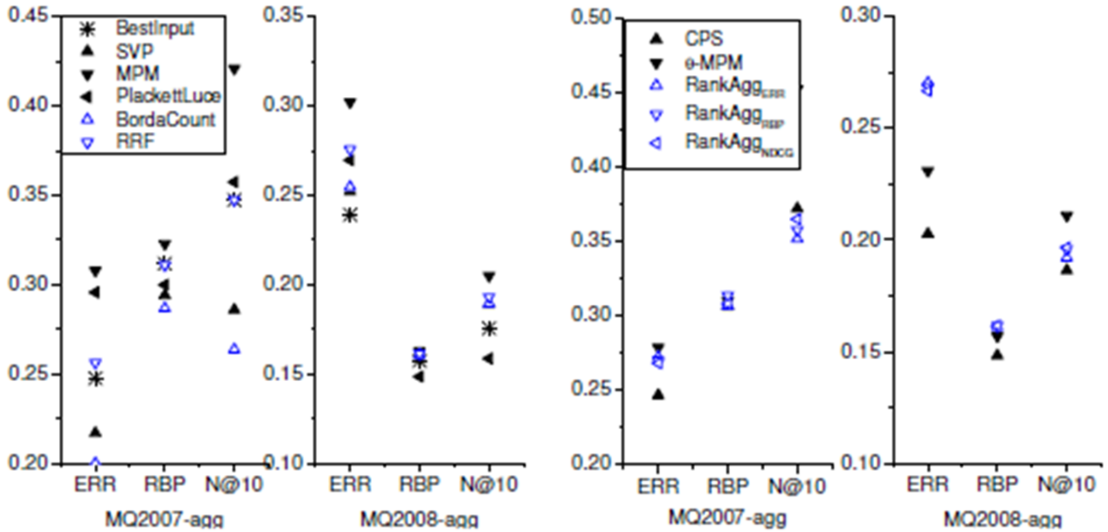
Rank Aggregation

Stochastic Rank Aggregation (UAI2013)
Listwise Approach for Rank Aggregation in CrowdSouring (WSDM2015)

Stochastic Rank Aggregation

Motivation

- Failure of explicit rank aggregation methods:



(a) Unsupervised scenario (b) Supervised scenario

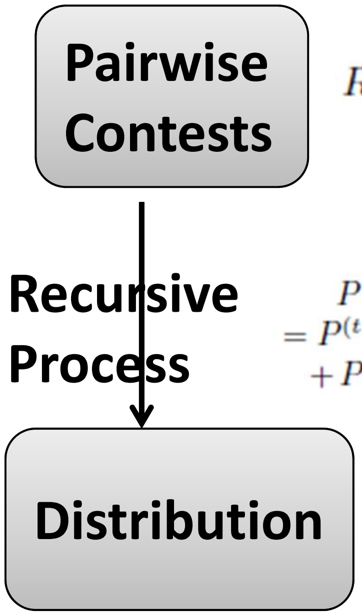
Explicit Methods Are Not So Good As Implicit Methods.

Unreliable Rank Information From Partial Ranking!

Incorporating Uncertainty into Rank Aggregation

Stochastic Rank Aggregation

A: Rank as A Random Variable

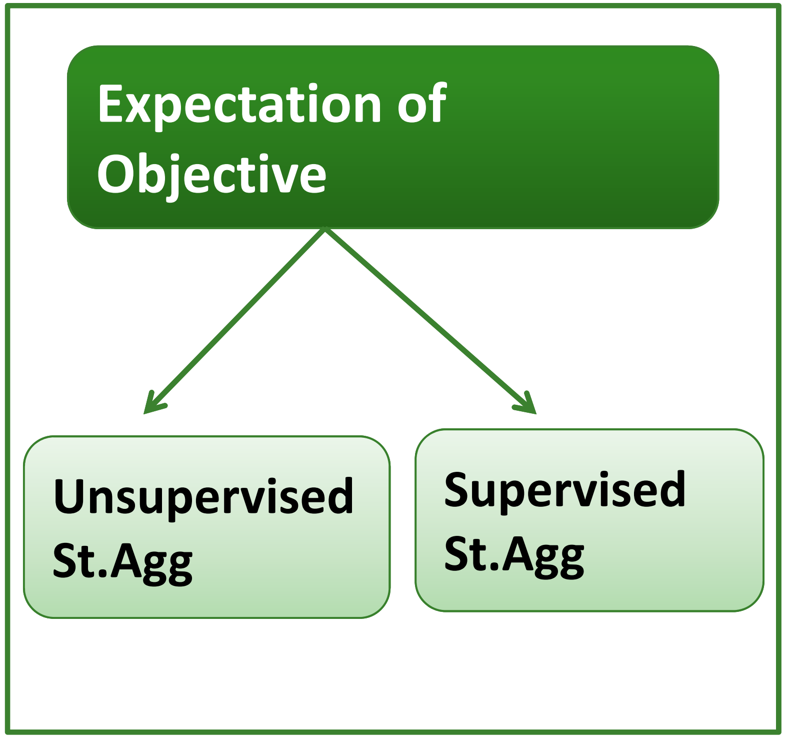


$$R(x_j,\tau) = \sum_{i=1, i\neq j}^{n} I(x_i \succ_\tau x_j),$$

$$\begin{aligned} & P^{(t)}(R(x_j,\tau)=r) \\ &= P^{(t-1)}(R(x_j,\tau)=r-1)p(x_i \succ_\tau x_j) \\ &\quad + P^{(t-1)}(R(x_j,\tau)=r)(1-p(x_i \succ_\tau x_j)). \end{aligned}$$

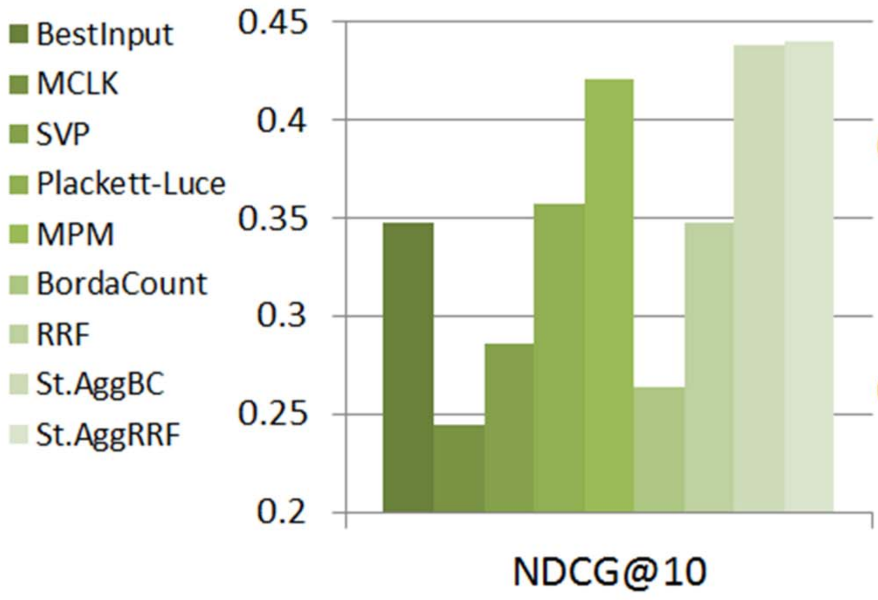
$$P(R(x_j,\tau)=r)$$

B: St.Agg Algorithm

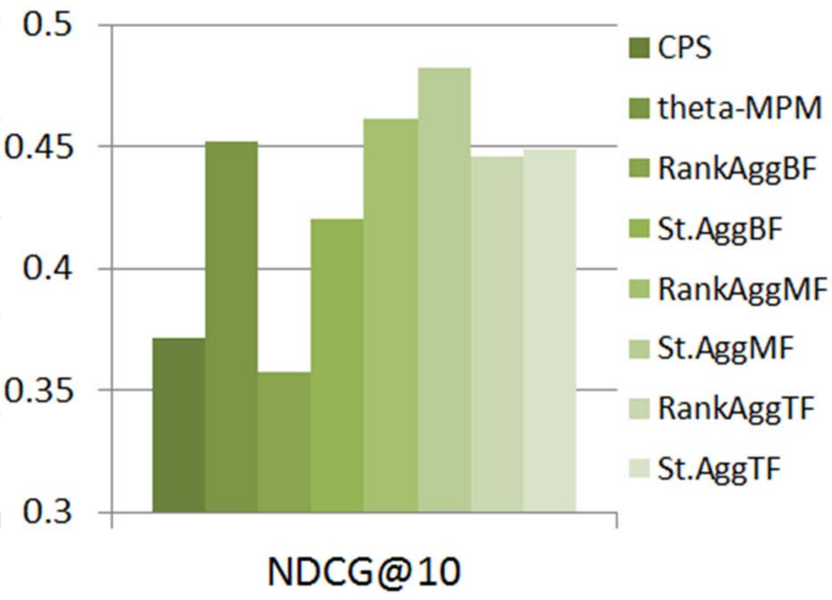


Experimental Results

- Metasearch data sets: MQ2007-agg and MQ2008-agg
- Effectiveness (e.g. MQ2007-agg)



Unsupervised



Supervised

Summary

- Beyond Relevance Ranking
  - Top-k Learning to Rank
  - Diverse Ranking: Relational Learning to Rank
  - Rank Aggregation
- Future Work
  - Learning to Match (Deep Matching)

Part III

Social media analytics

Part III

Social media analytics

- ✓ IMRank: Influence Maximization via Finding Self-Consistent Ranking (SIGIR 2014)
- ✓ StaticGreedy: Solving the Scalability-Accuracy Dilemma in Influence Maximization (CIKM 2013)
- ✓ Modeling and Predicting Popularity Dynamics via Reinforced Poisson Processes (AAAI 2014)
- ✓ Collective credit allocation in science (PNAS)
- ✓ Temporal scaling in information propagation (Sci. Rep.)
- ✓ Learning User-Specific Latent Influence and Susceptibility from Information Cascades (AAAI 2015)
- ✓ Context-Adaptive Matrix Factorization for Multi-Context Recommendation (CIKM 2015)
- ✓ Popularity prediction in microblogging network - a case study on Sina Weibo (WWW 2013)

Social media analytics: Outline

- Social influence
  - Influence maximization
  - User influence modeling
- Collective behavior
  - Popularity prediction
  - Credit allocation
- Sentiment classification

Social Media Analytics

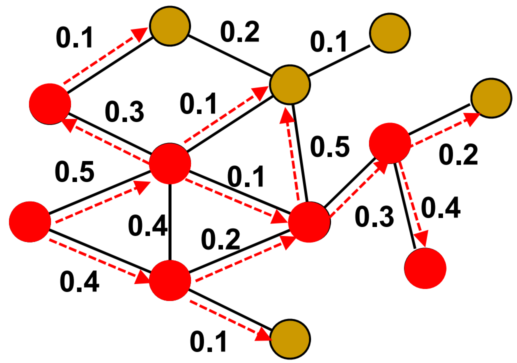
INFLUENCE MAXIMIZATION

- ✓ IMRank: Influence Maximization via Finding Self-Consistent Ranking (SIGIR 2014)
- ✓ StaticGreedy: Solving the Scalability-Accuracy Dilemma in Influence Maximization (CIKM 2013)

Influence maximization

- Finding a set of nodes to maximize the spread of influence in a given network

- Objective function
  - *Influence spread $I(S)$* : expected number of influenced nodes
  - Maximize $I(S)$
- Input:
  - A social influence graph $G=(V, E)$
  - *An information cascade model*
  - An integer k, $|S| \leq k$
- Output: A seed set S



Influence sparead

Cheng et al. CIKM 2013; Cheng et al., SIGIR 2014

Difficulties in Influence Maximization

Difficulty 1: Influence maximization problem is NP-hard.[kempe, KDD'0

Existing solutions

- **Heuristics**
  - **Degree**
  - **Pagerank**
  - **Betweennes**

efficient

inaccurate

- **Greedy approximate algorithm**
  [Kempe, KDD'03]
  - (1-1/e-ε)-approximation
  - iteratively select nodes with largest marginal influence spread
  - guaranteed by ***submodularity*** and ***montonicity*** properties of influence spread function

accurate

inefficient

Cheng et al. CIKM 2013; Cheng et al., SIGIR 2014

Difficulties in Influence Maximization

Difficulty 2: To exactly compute influence spread is #P-hard. [Chen, KDD'10]

Existing solutions

- Monte-Carlo simulation
  - CELF optimizatic
  - NewGreedy
  - CELF++ optimization[Goyal,WWW'11]

accurate

time-consuming

- ethods
  - scount[Che]
  - g, KDD'10]
  - PMIA[Chen,KDD'10]
  - IRIE[Jung, ICDM'12]

efficient

inaccurate

A scalability-accuracy dilemma!

Cheng et al. CIKM 2013; Cheng et al., SIGIR 2014

Our works

- Objective : to propose an influence maximization algorithm to *solve the scalability-accuracy dilemma*

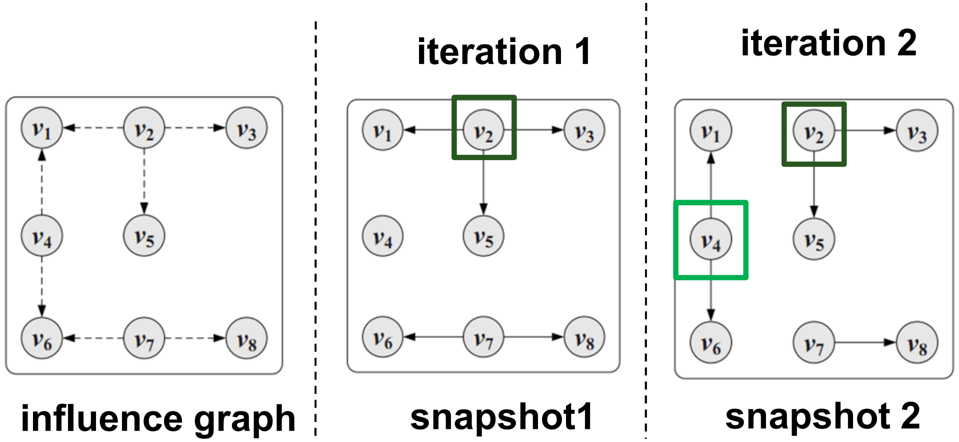
	Algorithm		Accuracy	Scalability
Approximate algorithms	Greedy	[Kempe, KDD'03]	**gurannteed**	**low**
	CreedyCELF	[Leskovec, KDD'07]	**gurannteed**	**low**
	GreedyCELF++	[Goyal, WWW'11]	**gurannteed**	**low**
	NewGreedy /MixedGreedy	[Chen, KDD'09]	**gurannteed**	**low**
	StaticGreedy	[cheng, CIKM'13]	***gurannteed***	***high***
Heuristics	Degree		**ungurannteed**	**high**
	PageRank	[Page, 1999]	**ungurannteed**	**high**
	DegreeDiscount	[Chen, KDD'09]	**ungurannteed**	**high**
	PMIA	[Chen, KDD'10]	**ungurannteed**	**high**
	IRIE	[Jung, ICDM'12]	**ungurannteed**	**high**
	SP1M	[Kimura, PKDD'06]	**ungurannteed**	**relatively low**

Cheng et al. CIKM 2013; Cheng et al., SIGIR 2014

Motivation

- Existing greedy algorithms
  - a risk of unguaranteed submodularity and monotonicity of influence spread function
    - **caused by using different results of Monte Carlo simulation across different influence spread estimation**
    - **a very large value of R is required, e.g. R=20000**

R: number of Monte Carlo simulations for estimation



$$I(S_0 \cup \{v_4\}) - I(S_0) = I(\{v_4\}) - I(\varnothing) = 1$$
$$I(S_1 \cup \{v_4\}) - I(S_1) = I(\{v_2, v_4\}) - I(\{v_2\}) = 3$$

Submodularity is breaked!

Cheng et al. CIKM 2013; Cheng et al., SIGIR 2014

StaticGreedy algorithm

- Core idea: to always use the same snapshots for influence spread estimation
  - influence spread function is submodular and monotone
  - a small value of R is required, e.g. R=100

Part1: Generate R static snapshots

Part 2: Greedy selection

Algorithm 1 StaticGreedy(G,k,R)

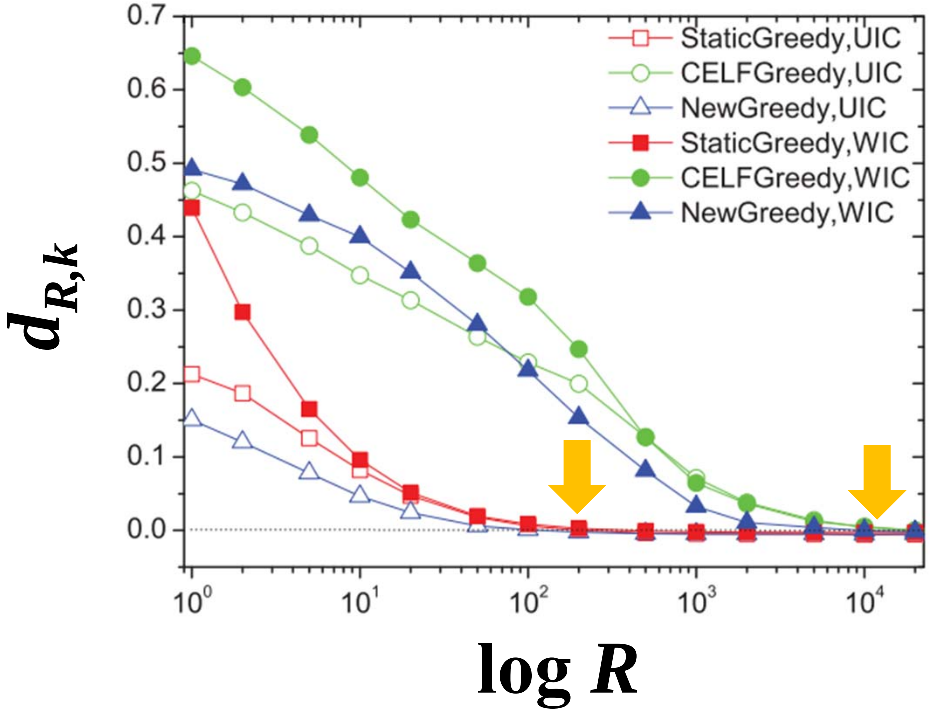
initialize $S = \varnothing$
for $i = 1$ to R **do**
  generate snapshot G'_i by removing each edge $\langle u, v\rangle$ from G with probability $1 - p(u, v)$
end for
for $i = 1$ to k **do**
  set s_v= 0 for all $v \in V \setminus S$ //s_v stores the influence spread after adding node v
  for $j = 1$ to R **do**
    for all $v \in V \setminus S$ **do**
      $s_v \mathrel{+}= |R(G'_j, S \cup \{v\})|$ //$R(G'_j, S \cup \{v\})$ is the influence spread of $S \cup \{v\}$ in snapshot G'_j
    end for
  end for
  $S = S \cup \{\arg\max_{v \in V \setminus S} \{s_v/R\}\}$
end for
output S

Performance analysis:
Convergence rate

- provide (1-1/e-ε)-approximation with a small value of R

$$d_{R,k} = \frac{I(S_k^*) - I(S_{R,k})}{I(S_k^*)}$$

seed set size = 50

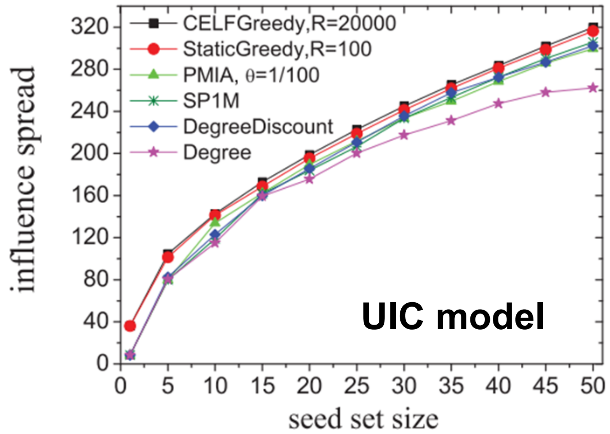


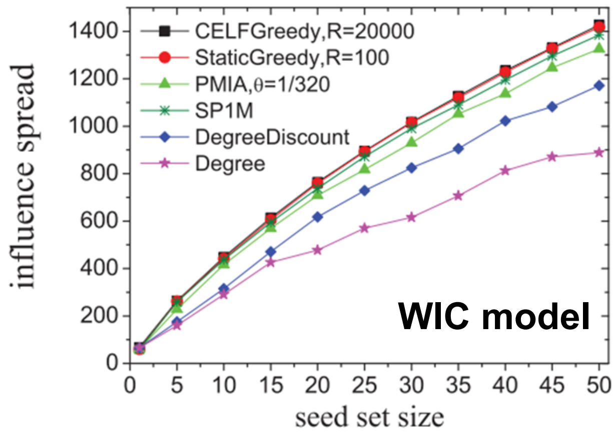
Cheng et al. CIKM 2013; Cheng et al., SIGIR 2014

Experiments: influence spread

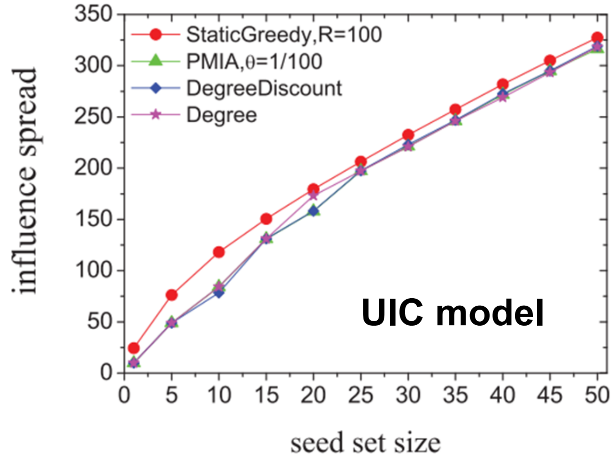
- StaticGreedy achieves better accuracy than other heuristics

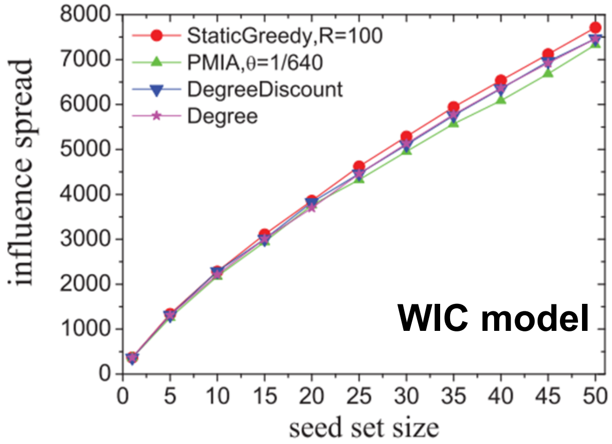
NetPHY





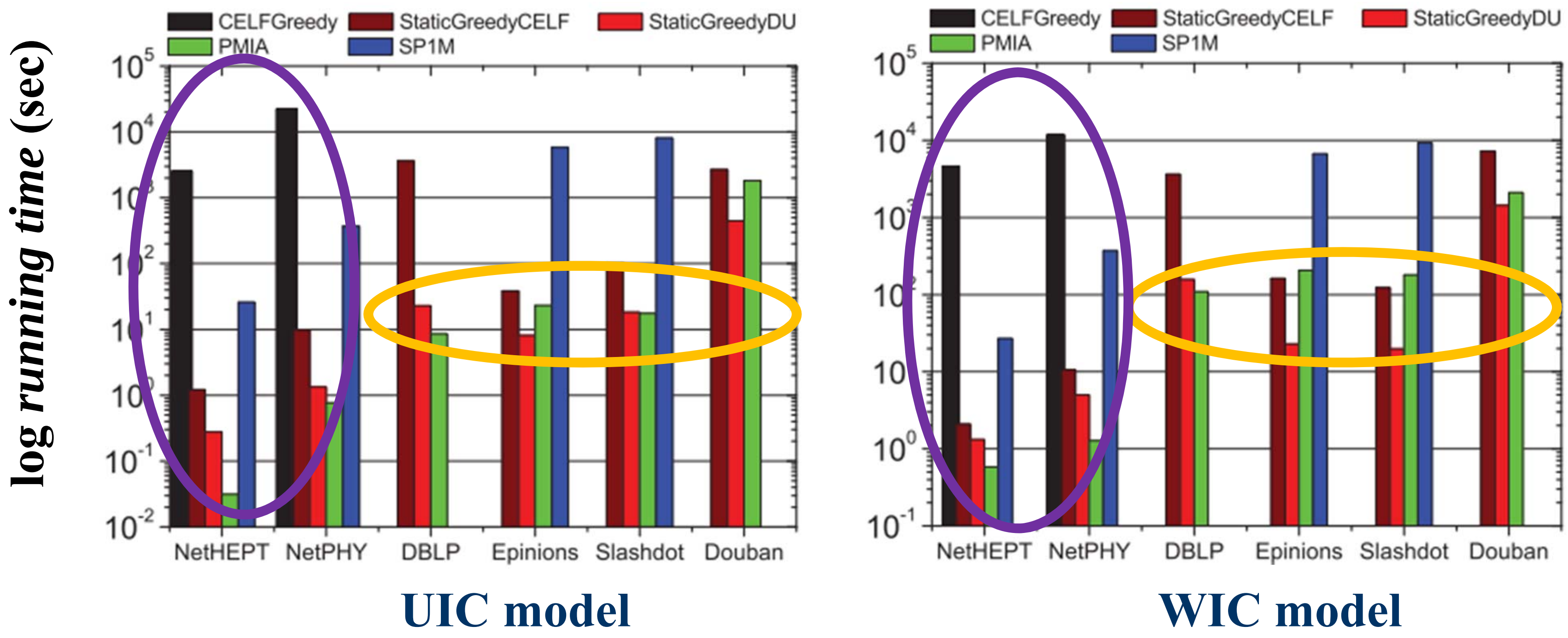
DBLP





Experiments: running time

- StaticGreedy runs **>10^3 times faster** than CELFGreedy
- StaticGreedy has **comparable scalability** to state-of-the-art heuristics
- StaticGreedyDU always runs faster than StaticGreedyCELF



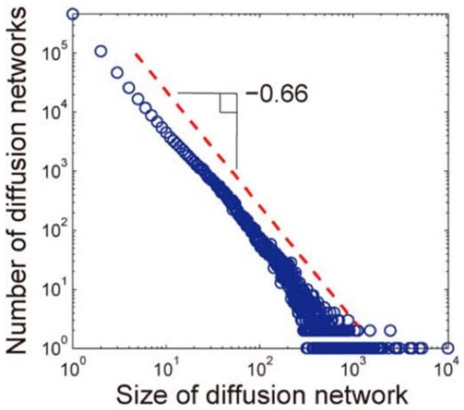
Social Media Analytics

POPULARITY PREDICTION

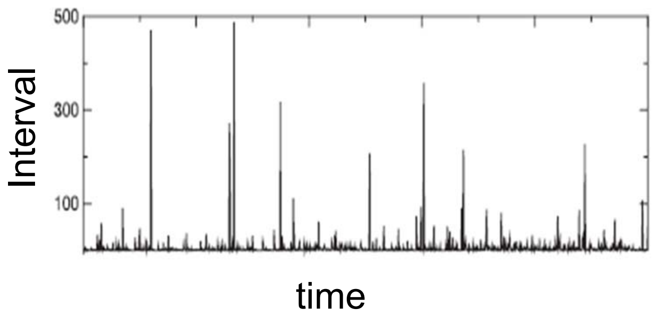
- ✓ Popularity prediction in microblogging network - a case study on Sina Weibo (WWW 2013)
- ✓ Modeling and Predicting Popularity Dynamics via Reinforced Poisson Processes (AAAI 2014)
- ✓ Learning User-Specific Latent Influence and Susceptibility from Information Cascades (AAAI 2015)

Popularity Prediction

- Challenges in Popularity Prediction

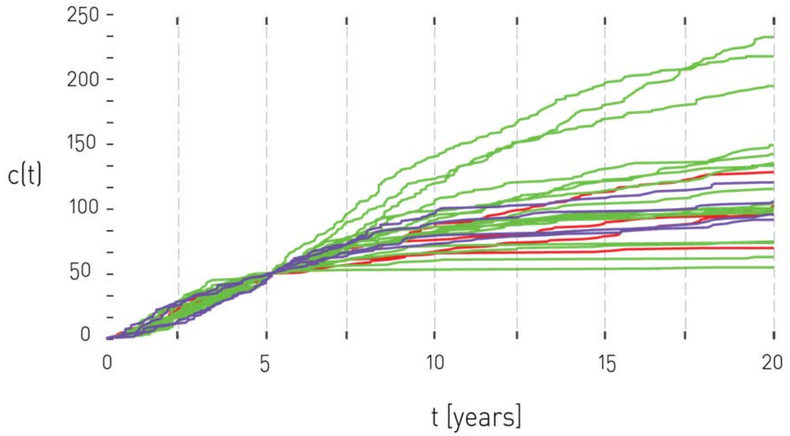


Imbalanced popularity distribution



Bursty

Citation count in early stage

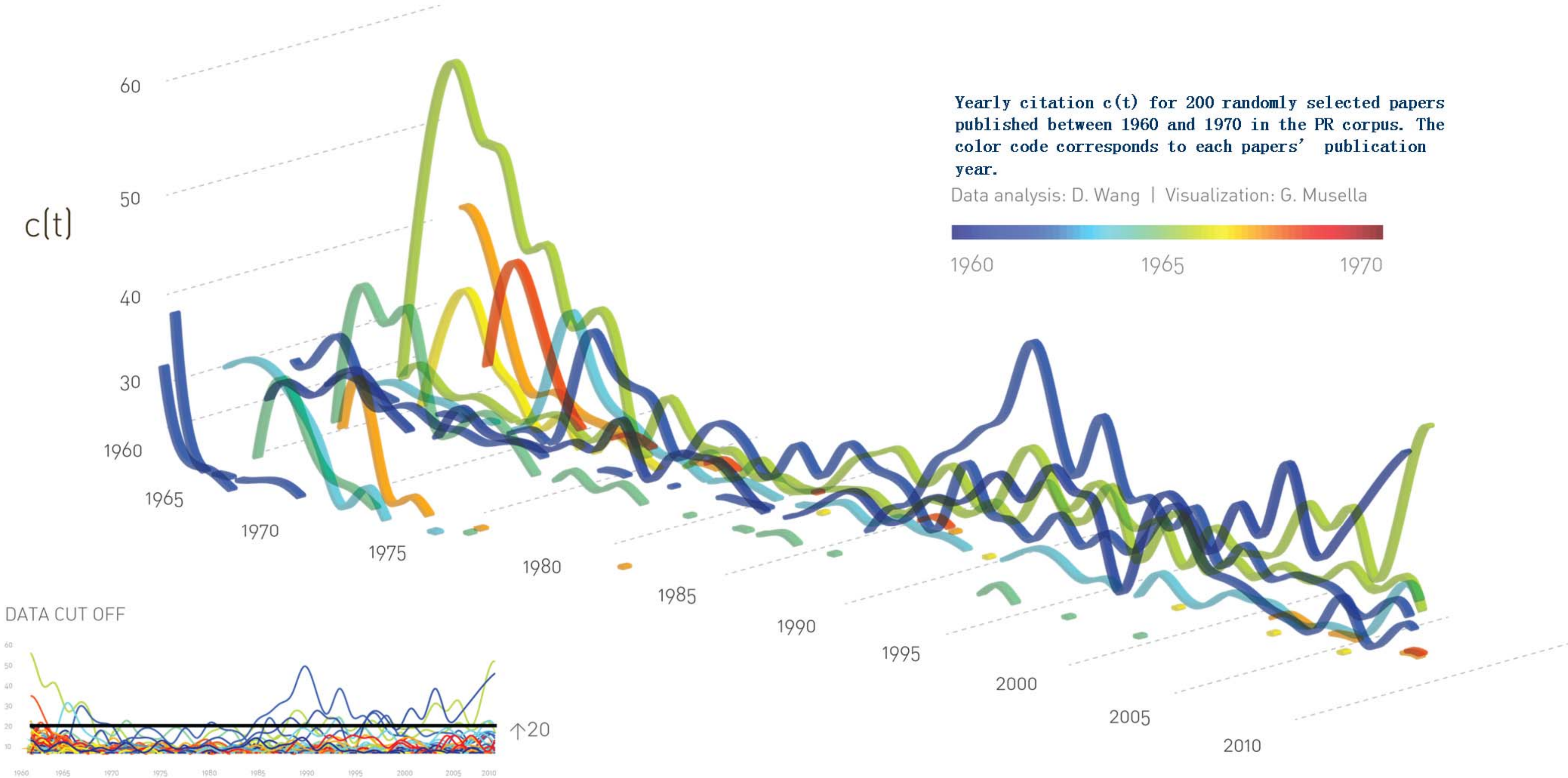


✓ Early popularity does not predict future popularity.

Bao et al., WWW 2013; Shen et al. AAAI 2014; Wang et al., AAAI 2015

Popularity Prediction

Heterogeneous popularity dynamics



Are the popularity dynamics predictable?

Bao et al., WWW 2013; Shen et al. AAAI 2014; Wang et al., AAAI 2015

Popularity Prediction

Modeling popularity dynamics

The rate of new attention to item i is : $\Pi_i \sim \eta_i c_i^t P_i(t)$

Intrinsic Novelty .... η_i

Preferential Attachment .... c_i^t

Aging effect .... $P_i(t) = \frac{1}{\sqrt{2\pi}\sigma_i t} \exp\left(-\frac{(\ln t - \mu_i)^2}{2\sigma_i^2}\right)$

$$\frac{\mathrm{d}c_i^t}{\mathrm{d}N} = \frac{\Pi_i}{\sum_{i=1}^{N} \Pi_i}$$

$$c_i^t = m\left(e^{\frac{\beta\eta_i}{A}\Phi\left(\frac{\ln t - \mu_i}{\sigma_i}\right)} - 1\right)$$

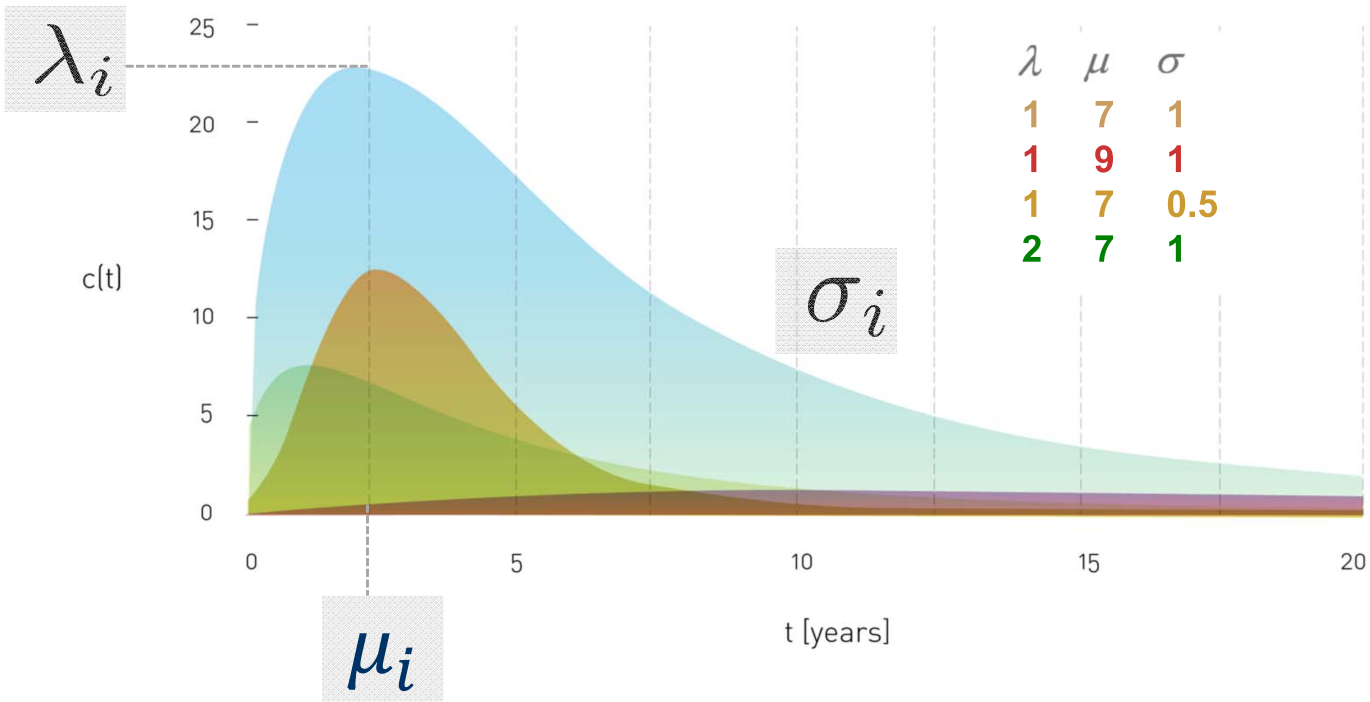
Bao et al., WWW 2013; Shen et al. AAAI 2014; Wang et al., AAAI 2015

Popularity Prediction

Physical meaning of parameters:

Popularity dynamics:

$$c_i^t = m\left(e^{\lambda_i \Phi\left(\frac{\ln t - \mu_i}{\sigma_i}\right)} - 1\right)$$



Fitness λ_i

Immediacy μ_i

Longevity σ_i

Bao et al., WWW 2013; Shen et al. AAAI 2014; Wang et al., AAAI 2015

Popularity Prediction

$$i_d(t) = m + i - 1$$

Generative model of popularity dynamics:

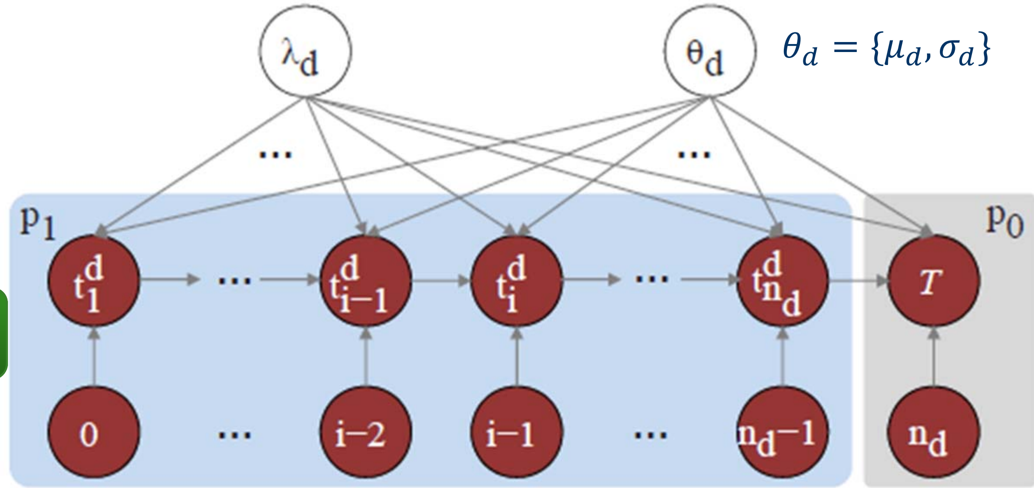
Reinforced Poisson Process:

$$x_d(t) = \lambda_d f_d(t;\theta_d) i_d(t)$$

Fitness

Aging effect

Rich gets richer



MLE for parameter estimation:

$$\begin{aligned}\mathcal{L}(\lambda_d,\theta_d) &= p_0(T|t_{n_d}^d)\prod_{i=1}^{n_d} p_1(t_i^d|t_{i-1}^d)\\ &= \lambda_d^{n_d}\prod_{i=1}^{n_d}(m+i-1)f_d(t_i^d;\theta_d)\times\\ &\quad e^{-\lambda_d\left((m+n_d)F_d(T;\theta_d)-\sum_{i=1}^{n_d}F_d(t_i^d;\theta_d)\right)}\end{aligned}$$

Prediction:

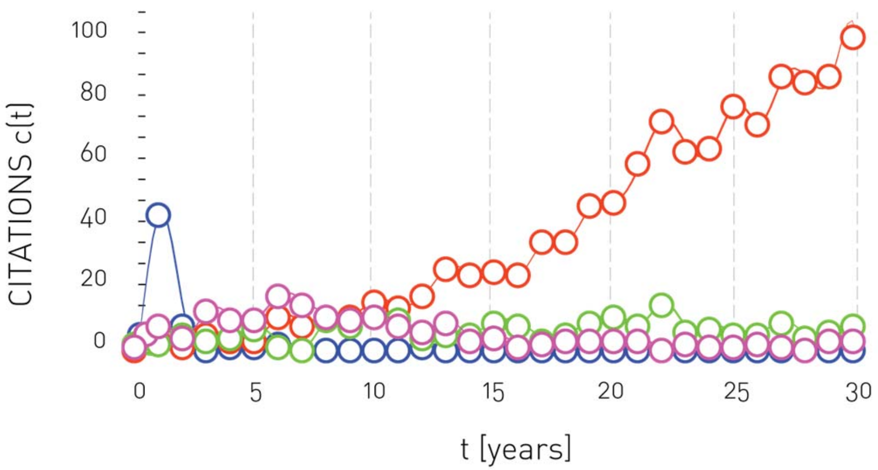
$$\frac{\mathrm{d}c^d(t)}{\mathrm{d}t} = \lambda_d f_d(t;\theta_d)(m + c^d(t))$$

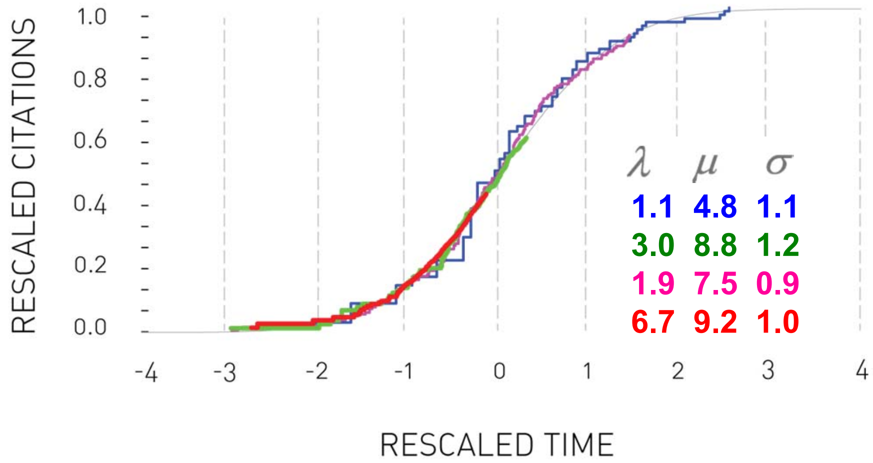
$$c^d(t) = (m+n_d)e^{\lambda_d^*\left(F_d(t;\theta_d^*)-F_d(T;\theta_d^*)\right)} - m$$

Bao et al., WWW 2013; Shen et al. AAAI 2014; Wang et al., AAAI 2015

Popularity Prediction

Examples:

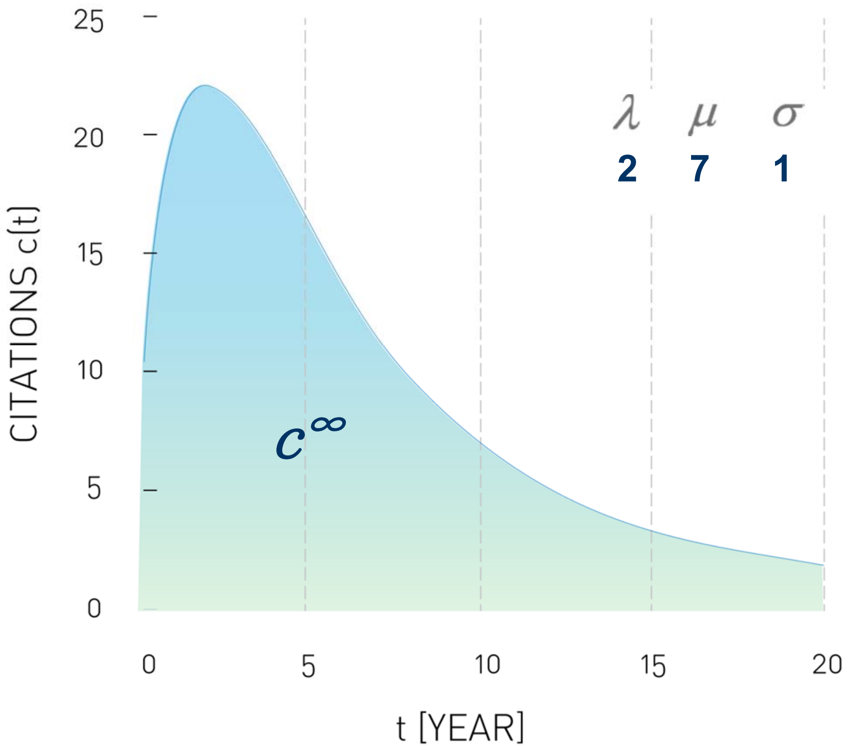




$$\tilde{t} \equiv (\ln t - \mu_i)/\sigma_i$$
$$\tilde{c} \equiv \ln(1 + c_i^t/m)/\lambda_i \longrightarrow \tilde{c} = \Phi\left(\tilde{t}\right)$$

Bonner & Fisher, *Linear magnetic chains with anisotropic coupling,* Physical Review (1964)
Hohenberg & Kohn, *Inhomogeneous electron gas,* Physical Review (1964)
Bardakci et al. *Intrinsically Broken U(6) ⊗ U(6) Symmetry for Strong Interactions,* Physical Review Letters (1964)
Berglund & W.E. Spicer, *Photoemission studies of copper and silver: Theory,* Physical Review (1964)

Popularity Prediction



The final popularity c^∞ of an item is

$$c_i^t = m\left(e^{\lambda_i \Phi\left(\frac{\ln t - \mu_i}{\sigma_i} - 1\right)}\right)$$

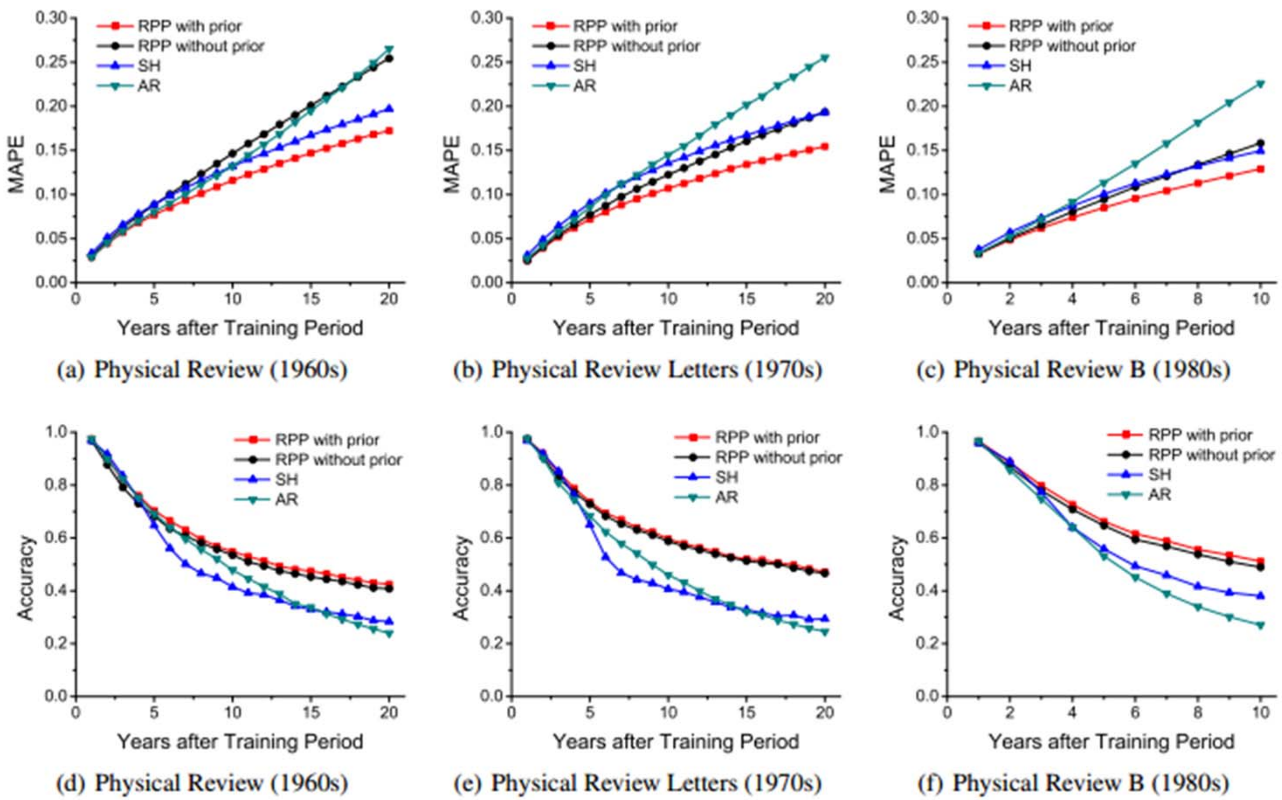
$$\boxed{c_i^\infty = m(e^{\lambda_i} - 1)}$$

- ✓ **Final popularity depends only on a the attractiveness of item**

Bao et al., WWW 2013; Shen et al. AAAI 2014; Wang et al., AAAI 2015

Popularity Prediction

Results – Citation Count Prediction



(a) Physical Review (1960s) (b) Physical Review Letters (1970s) (c) Physical Review B (1980s)

(d) Physical Review (1960s) (e) Physical Review Letters (1970s) (f) Physical Review B (1980s)

Data:

- 463 348 papers
- 4 710 547 citations
- American Physical Society (1893-2009)

✓ RPP (Reinforced Poisson Process) consistently outperforms competing methods.

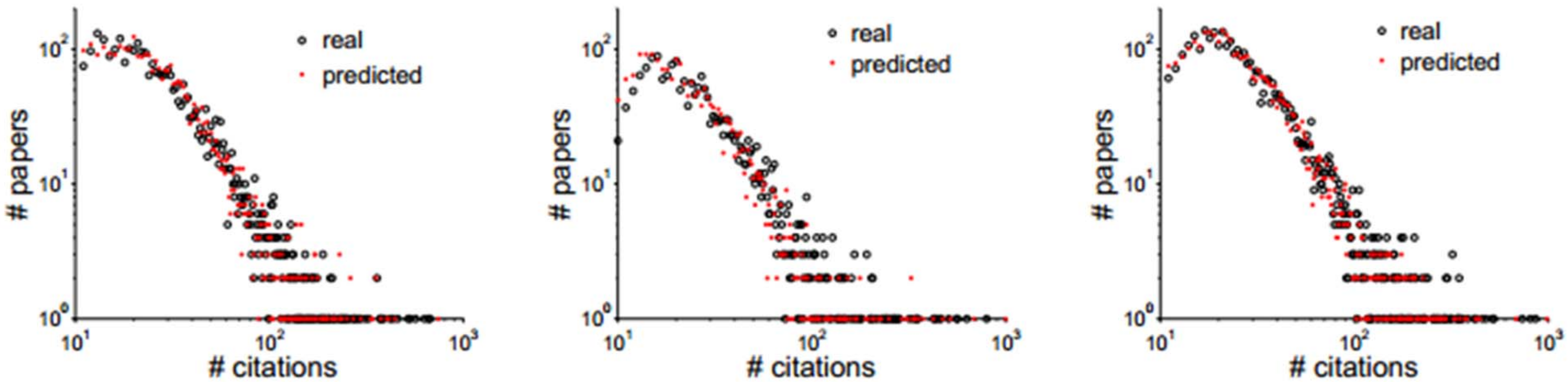
✓ RPP without prior performs almost identically to RPP with prior (high accuracy), but performs remarkably bad on a handful of cases, caused by overfitting (high MAPE)

✓ The superiority of the RPP with prior, increases with the length of training periods.

Bao et al., WWW 2013; Shen et al. AAAI 2014; Wang et al., AAAI 2015

Popularity Prediction

Results – Citation Count Prediction



(g) Physical Review (1960s) (h) Physical Review Letters (1970s) (i) Physical Review B (1980s)

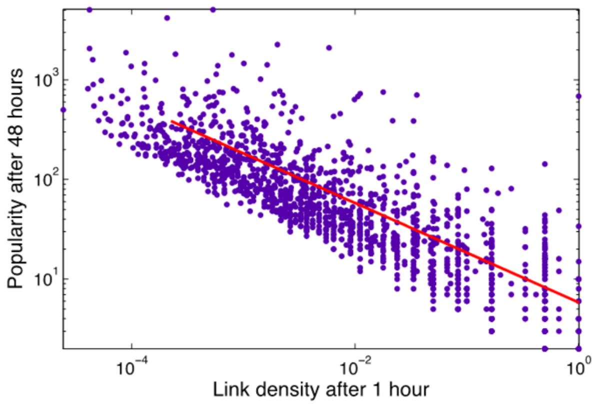
- ✓ **The RPP model is able to reproduce the citation distribution, indicating that the RPP model can also be used to model the global properties of citation system.**

Bao et al., WWW 2013; Shen et al. AAAI 2014; Wang et al., AAAI 2015

Popularity Prediction

Results – Weibo Retweet Count Prediction

- Data
  - Sina Weibo，July 1-31, 2011, 16.6M messages
  - Incorporating structural features



Link density vs. Popularity

Primitive type	RMSE	MAE
Baseline	0.77	0.57
with link density	0.63	0.45
with diffusion depth	**0.61**	**0.43**

20% reduction of error

Bao et al., WWW 2013; Shen et al. AAAI 2014; Wang et al., AAAI 2015

Popularity Prediction

Summary

- Popularity dynamics follow a universal law, incorporating three mechanisms
  - Survival of the fittest
  - Rich gets richer
  - Aging factor
- The arriving process of citations is modeled via reinforced Poisson process
  - Instead of time series or aggregated curve fitting
- Working in the manner of probabilistic generative model
  - Flexible to incorporate prior, providing higher predictive power
  - A kernel-style relaxation function is used to model aging factor, providing the possibility to be adapted according to contexts, e.g., microblogging

Social Media Analytics

SENTIMENT CLASSIFICATION

- ✓ **Adaptive Co-Training SVM for Sentiment Classification on Tweets, CIKM 2013**
- ✓ **Co-training and Visualizing Sentiment Evolvement for Tweet Events, WWW 2013**
- ✓ **SUIT: A Supervised User-Item based Topic model for Sentiment Analysis, AAAI 2014**
- ✓ **TASC: Topic-Adaptive Sentiment Classification on Dynamic Tweets, TKDE 2015**

Sentiment Analysis

- Opinion information is Important
  - Individual Consumer
    - Make a better decision when buying products
  - Business Company
    - Product improvement
    - Marketing strategy
- Sentiment classifiers dedicates to a specific topic
  - The same *word* for different topics may have different sentiment orientations

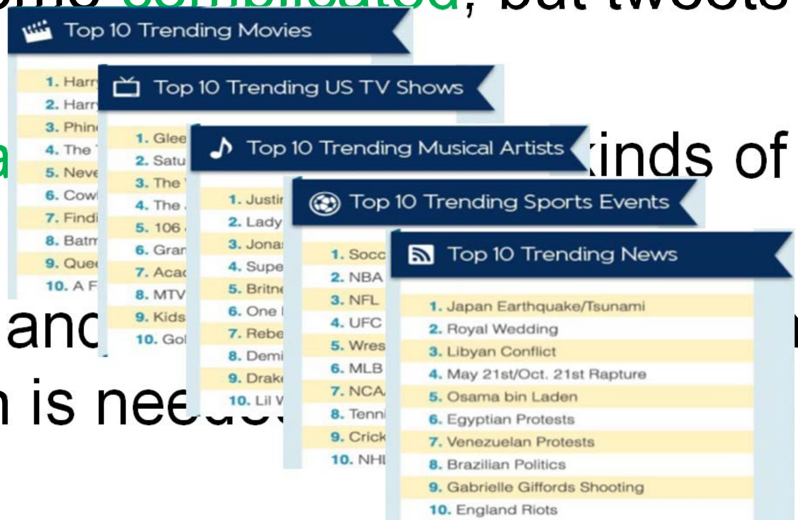
e.g. "Long"

Positive* *for cellphone battery

Negative* *for camera focus time

Need Adaptive and Semi-supervised

- Topics in twitter are more diverse
- Emoticons in tweets were ever used as noisy labels
  - haven't become a convention
  - neutral class could not be labeled
- Models become complicated, but tweets lack sufficient labeled data
- Giving pre-la ... kinds of topics is impossible.
- An adaptive and ... ment classification is nee...

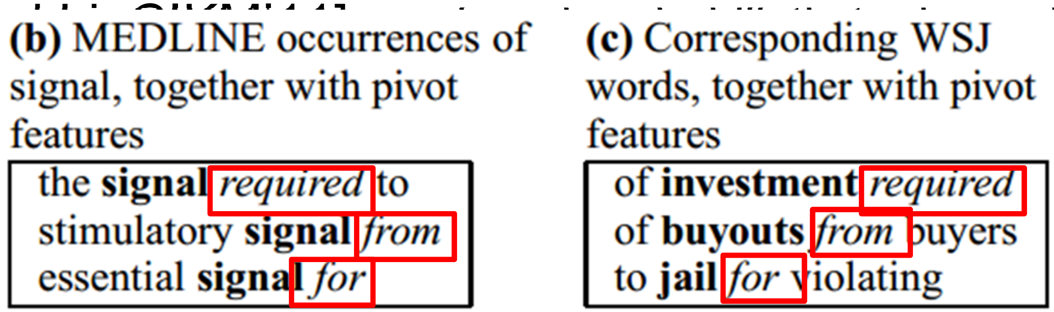


Previous Works

- PMI-IR [Turney, ACL'02] proposed an web-kernal based PMI (Pointwise mutual information) for unsupervised sentiment classification.
  - Discard some supervised information
  - Cannot dive into topic-specific sentiment features
- SCL [Blitzer *et al,* ACL'07] explicitly borrowed a bridge to connect the topic-dependent features to a known or common features.

[Gao ar ... el to bridge different doı

SFA [P ... between topic-independen ... g to co-align those words betw



  - They assumed that the parallel sentiment words exists for each pair of topics
  - Twitter contains more diversified topics, and are unknown before classification.

Our Adaptive and Semi-supervised Solution

- With
  - a small amount of supervised information
  - Topic-independent features: *sentiment words (PMI-IR), emoticons, post times, punctuations etc*.
- Iteration
  - Adapting to unlabeled data on a target topic in transductive way
  - Adapting to the topic-specific words. [Liu, et al, CIKM'13]
  - Adapting to user-level and network-based features. [Liu, et al, TKDE'15]
- Key to topic-adaptive sentiment classification
  - Extract and estimate sentiment polarity of topic-specific words

Observations on Users' opinions

- User's opinion on a topic is consistent

Sentiment Statistics of Users' Tweets

	total users	users (≥2 tweets)	average var
Taco Bell	3,446	106	0.1008
President Debate	1,204	520	0.4168

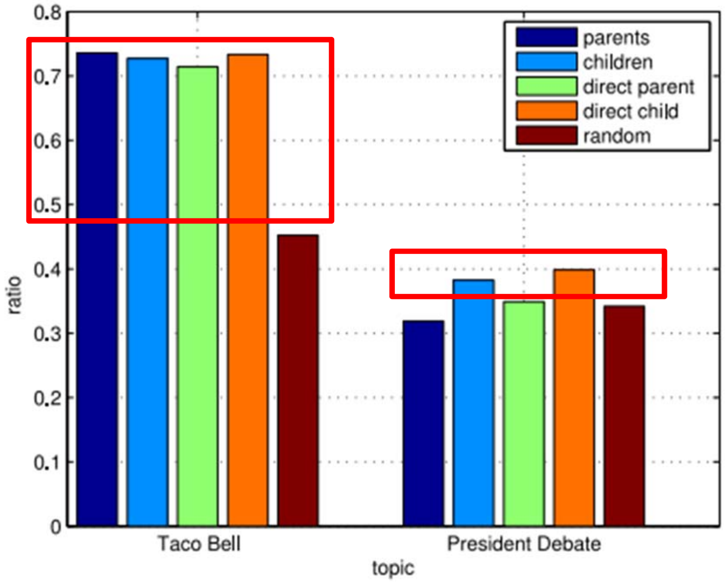
< 0.67 (random)

Example

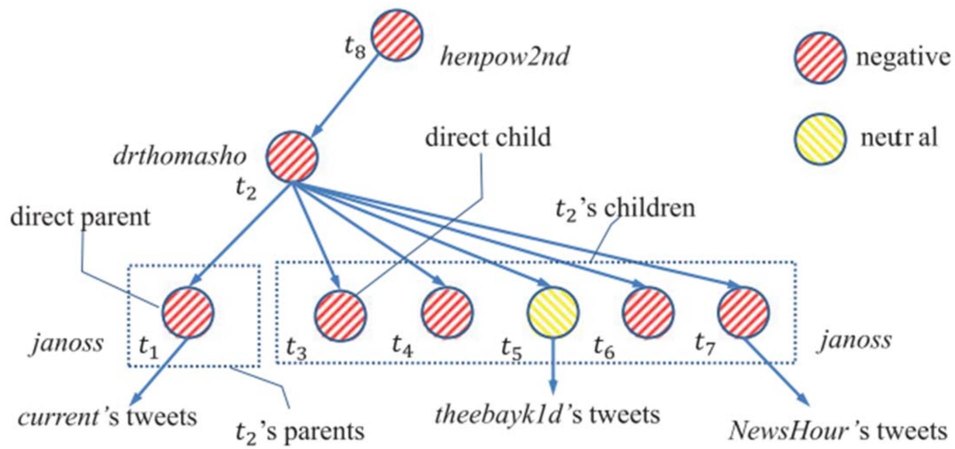
Content	user
+1 for Obama "Moving fast, moving swiftly" #tweetdebate	nohype
Obama +1 pt: We need more responsibility but not just during a crisis. #tweetdebate	nohype
+2 to Twitter for handling this so well (so far). #tweetdebate	nohype
Obama won McCain just rambled #current	nohype

Observations on Users' @-network

- Herd effect in opinions of users in a @-network (mention each other)



Example



	post time	author	@ whom	sentiment
t_1	2:01	*janoss*	@current	negative
t_2	2:13	*drthomasho*	@janoss	negative
t_3	2:19	*janoss*	--	negative
t_4	2:29	*janoss*	--	negative
t_5	2:37	*janoss*	@theebayk1d	neutral
t_6	2:38	*janoss*	--	negative
t_7	2:55	*janoss*	@NewsHour	negative
t_8	2:57	*henpow2nd*	@drthomasho	negative

TASC: Topic-Adaptive Sentiment Classification

- Features in TASC model

$$x = \overbrace{x_1, x_2, \ldots, x_v, \underbrace{x_{v+1}, \ldots, x_{v+u}}_{\text{Topic-specific words}}}^{\text{Text features}}, \underbrace{x_{v+u+1}, \ldots, \underbrace{x_{v,1\sim K}}_{\text{User-level features}}, \underbrace{x_{\rho,1\sim K}, x_{\sigma,1\sim K}}_{\text{Network-based features}}}_{\text{Non-text features}}$$

S. Liu, F. Li et al., CIMK 2013, S. Liu, X. Cheng et al., TKDE 2015

TASC model

- A unified TASC model

$$y' = \arg\max_{y}\{f(x,\omega_y)\}$$

$$\min_{x_{v+1},\cdots,x_{v+w},\omega} \underbrace{L(f(x,\omega),y) + L'(f(x,\omega),\boxed{y'})}_{\textbf{Loss funcitons}} + \underbrace{R(\omega)}_{\textbf{regularization/prior}}$$

  - Semi-supervised: minimize the loss of unlabeled data L'(·)
  - Topic-adaptive: topic-specific features as optimized variables

An Instance of TASC model

Choose linear function, logistic loss and L_2 regularization

$$\min_{\theta, x_{v+1\sim v+u}} -\frac{C}{|L|}\sum_{t_i \in L}\sum_{j=1}^{K}[1 - \mathbf{1}(y_i = j)]\log(y_i = j|x_i;\theta)) - \frac{C'}{|U|}\sum_{t_i \in U}\sum_{j=1}^{K}[1 - \mathbf{1}(y_i = j)]\log\big(P(y_i' = j|x_i;\theta)\big) + \frac{1}{2}\sum_{j=1}^{K}\theta_i^T\theta_i$$

Labeled data

Unlabeled data

For unlabeled tweet t_i, sentiment label is $y_i' = \arg\max_y \{w_y^T x_i\}$

where

- $\mathbf{1}(y^{(i)} = j)$ **is an indicator function. r**
- **Probability of tweet with feature** x belonging to class j

$$P(y = j|x;\theta) = \frac{\exp(\theta_j^T \cdot x)}{\sum_{1\le i\le k}\exp(\theta_i^T \cdot x)}$$

Experiments

- Test cases of 3 publicly available corpuses

Topics	Positive	Neutral	Negative	Total
Apple	191	581	377	1149
Google	218	604	61	883
Microsoft	93	671	138	902
Twitter	68	647	78	793
Taco Bell	902	2099	596	3597
President Debate	1465	1019	729	3213

  - Sanders-Twitter Sentiment Corpus
  - Taco Bell Corpus
  - The first 2008 Presidential Debate Corpus

- Baselines
  - DT: *Decision Tree*, MSVM: *multiclass SVM*, RF: *Random Forest*
  - MS3VM: *Semi-supervised SVM*, CoMS3VM: *MS3VM in co-training scheme*.

Comparisons with baselines

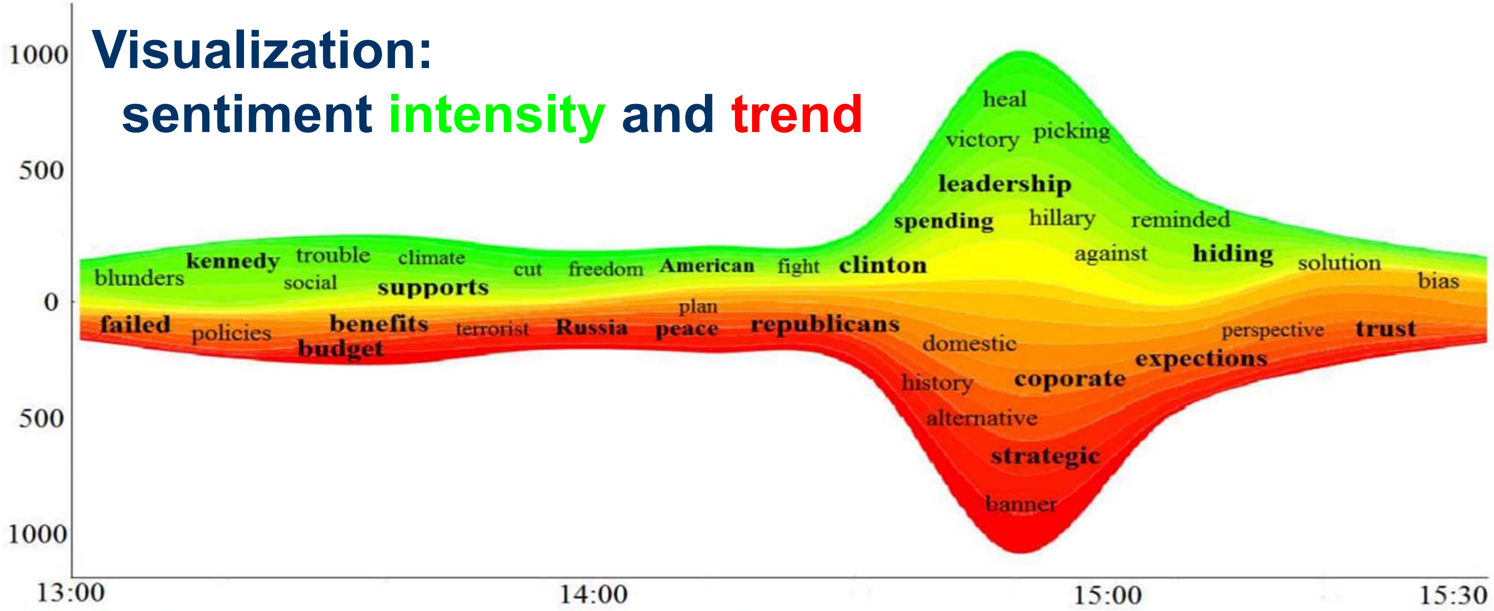
Comparisons with Baselines in 10% Sample Ratio

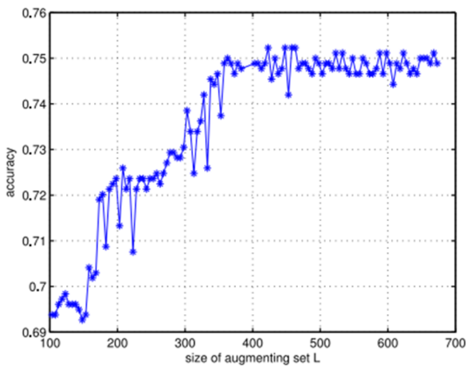
Topics	DT		MSVM		RF		MS3VM		CoMS3VM		TASC		TASC-t	
	Acc	F-s.	Acc.	F-s.	Acc.	F-s.	Acc.	F-s.	Acc.	F-s.	Acc	F-s.	Acc	F-s.
Apple	0.5063 ±0.0000	0.3400 ±0.0000	0.5036 ±0.0106	0.4624 ±0.0085	0.5403 ±0.0161	0.5063 ±0.0123	0.5116 ±0.0205	0.4344 ±0.0283	0.6795 ±0.0126	0.6440 ±0.0133	**0.6882** ±0.0114	**0.6528** ±0.0123	0.5461	0.4617
Google	0.6835 ±0.0000	0.5550 ±0.0000	0.7016 ±0.0068	0.6386 ±0.0141	0.7614 ±0.0255	**0.7414** ±0.0301	0.7614 ±0.0168	0.6350 ±0.0142	0.7662 ±0.0124	0.6201 ±0.0200	**0.7725** ±0.0119	0.6371 ±0.0218	0.7054	0.5155
Microsoft	0.7429 ±0.0000	0.6330 ±0.0000	0.7300 ±0.0186	0.6716 ±0.0057	0.7411 ±0.0123	**0.6894** ±0.0156	0.7315 ±0.0089	0.4355 ±0.0141	0.7884 ±0.0171	0.6058 ±0.0400	**0.7896** ±0.0176	0.6072 ±0.0363	0.7416	0.4809
Twitter	0.8112 ±0.0000	0.7270 ±0.0000	0.7976 ±0.0021	0.7260 ±0.0020	0.8097 ±0.0148	**0.7645** ±0.0065	0.8054 ±0.0086	0.5226 ±0.0108	0.8126 ±0.0157	0.5343 ±0.0225	**0.8176** ±0.0165	0.5472 ±0.0240	0.8196	0.4867
Taco Bell	0.5836 ±0.0000	0.4300 ±0.0000	0.6500 ±0.0075	0.5796 ±0.0120	0.5976 ±0.0088	0.5744 ±0.0073	0.6974 ±0.0240	0.5911 ±0.0463	0.7105 ±0.0034	0.6181 ±0.0077	**0.7126** ±0.0015	**0.6206** ±0.0058	0.7151	0.6297
President Debate	0.3845 ±0.0047	0.2422 ±0.0624	0.4365 ±0.0081	0.4228 ±0.0057	0.4848 ±0.0103	0.4858 ±0.0107	0.5167 ±0.0289	**0.5189** ±0.0212	0.5185 ±0.0287	0.5162 ±0.0240	**0.5216** ±0.0289	0.5175 ±0.0246	0.5901	0.5824

- TASC outperforms other baselines in mean accuracies on all the topics.

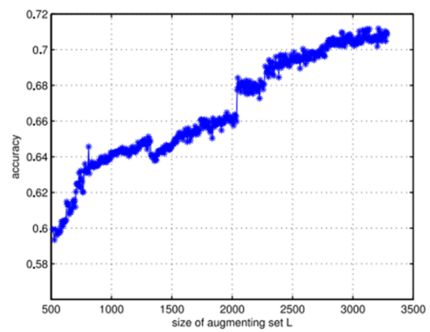
Experiments

Visualization:
sentiment intensity and trend

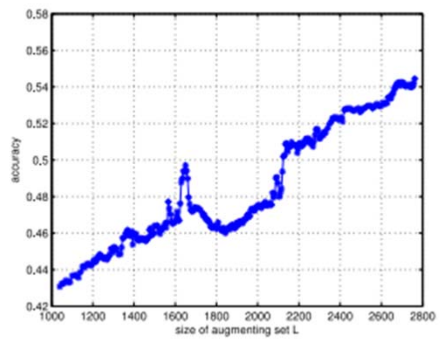




(a) sample 1% on Google.



(e) sample 5% on Taco Bell.



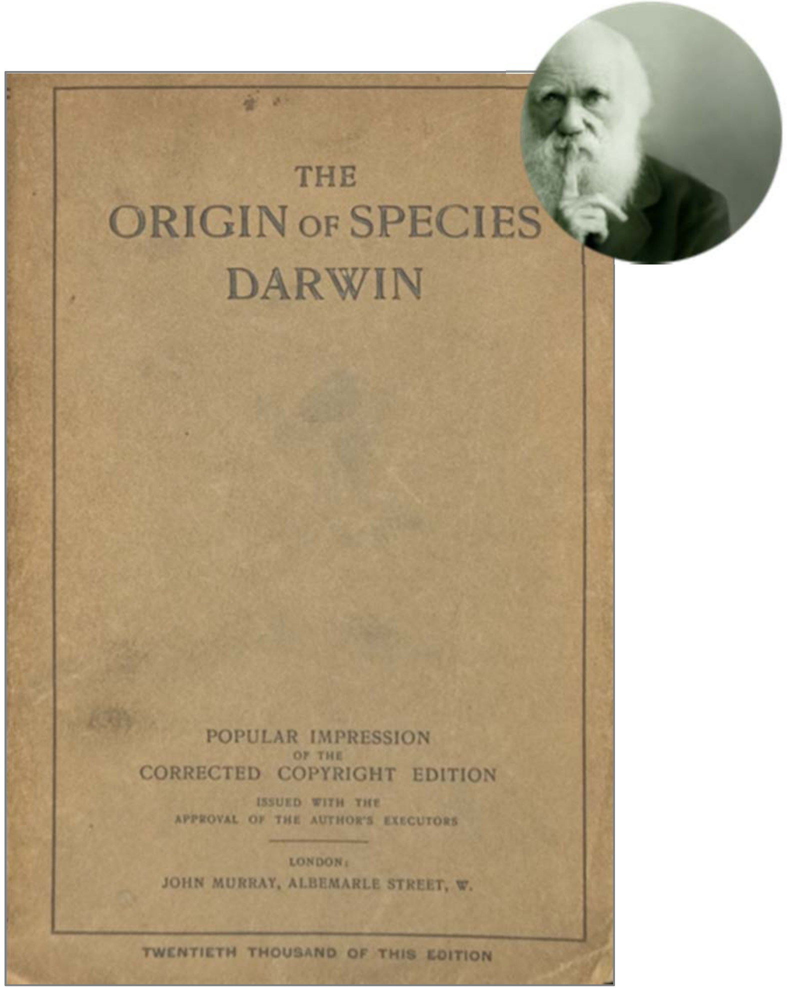
(i) sample 10% on President Debate.

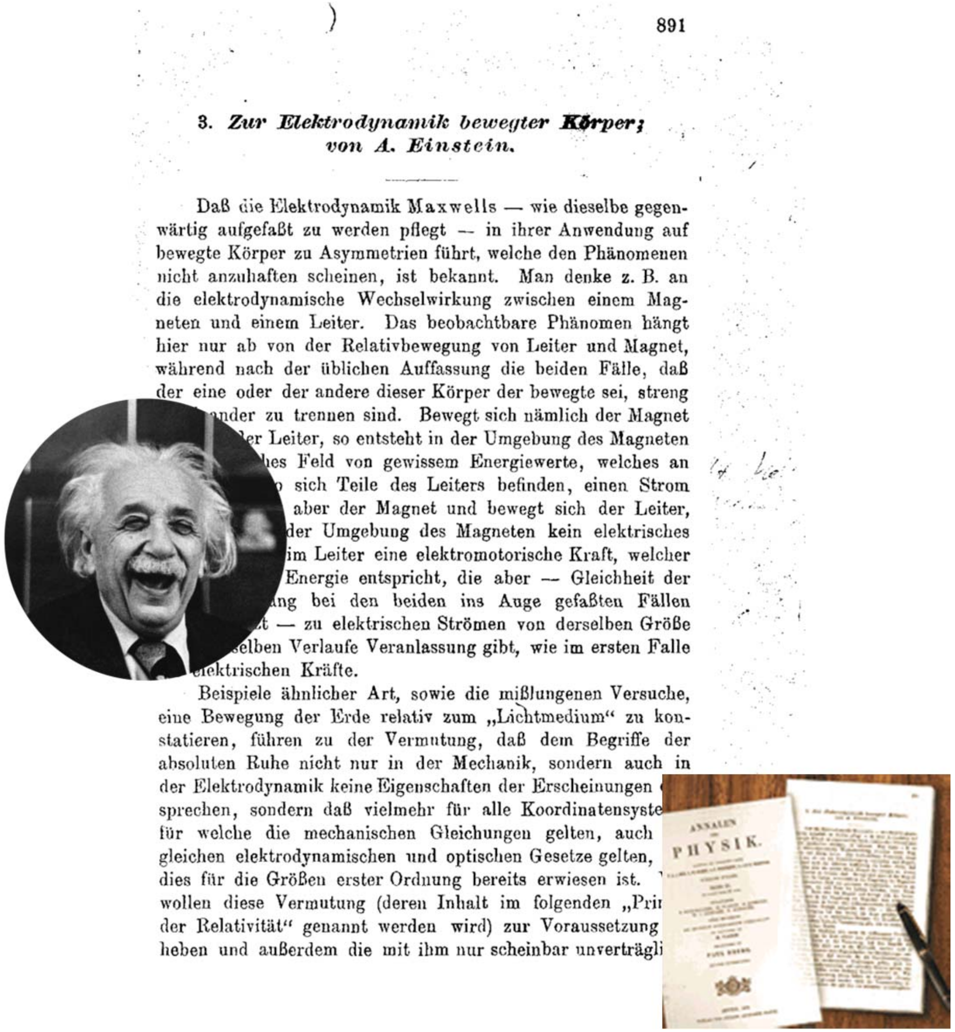
Social Media Analytics

CREDIT ALLOCATION

Shen et al., Collective credit allocation in science, PNAS, 2014.

Credit allocation



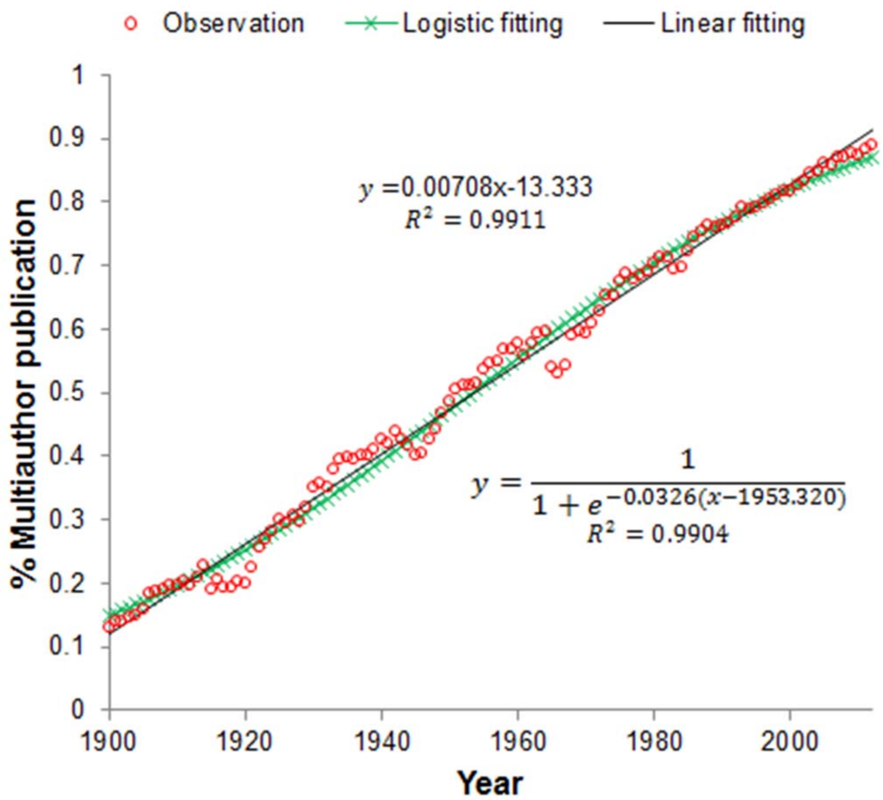


Simple rule of credit allocation:

the sole author gets all the credit for his discovery.

Shen & Barabási, PNAS, 2014

Credit allocation



Multi-author papers are dominating the publication of science, increasing by 7 percent every 10 years between 1900 and 2012.

Science's credit system is under pressure to evolve:
The norm of credit allocation for single-author publications fails for multi-author publications.

Shen & Barabási, PNAS, 2014

Credit allocation

2012 Nobel Prize-winning paper in Physics

VOLUME 76, NUMBER 11 PHYSICAL REVIEW LETTERS 11 MARCH 1996

Generation of Nonclassical Motional States of a Trapped Atom

D. M. Meekhof, C. Monroe, B. E. King, W. M. Itano, and [redacted]
Time and Frequency Division, National Institute of Standards and Technology, Boulder, Colorado 80303-3328

1997 Nobel Prize-winning paper in Physics

VOLUME 55, NUMBER 1 PHYSICAL REVIEW LETTERS 1 JULY 1985

Three-Dimensional Viscous Confinement and Cooling of Atoms by Resonance Radiation Pressure

[redacted] L. Hollberg, J. E. Bjorkholm, Alex Cable, and A. Ashkin
AT&T Bell Laboratories, Holmdel, New Jersey 07733

2007 Nobel Prize-winning paper in Physics

VOLUME 61, NUMBER 21 PHYSICAL REVIEW LETTERS 21 NOVEMBER 1988

Giant Magnetoresistance of (001) Fe/(001) Cr Magnetic Superlattices

M. N. Baibich,[a] J. M. Broto, [redacted] F. Nguyen Van Dau, and F. Petroff
Laboratoire de Physique des Solides, Université Paris-Sud, F-91405 Orsay, France

P. Eitenne, G. Creuzet, A. Friederich, and J. Chazelas
Laboratoire Central de Recherches, Thomson CSF, B.P. 10, F-91401 Orsay, France
(Received 24 August 1988)

Shen & Barabási, PNAS, 2014

Credit allocation

Volume 122B, number 1 PHYSICS LETTERS 24 February 1983

1984 Nobel Prize-winning paper in Physics

EXPERIMENTAL OBSERVATION OF ISOLATED LARGE TRANSVERSE ENERGY ELECTRONS WITH ASSOCIATED MISSING ENERGY AT $\sqrt{s}$ = 540 GeV

UA1 Collaboration, CERN, Geneva, Switzerland

Alphabetic author list.

G. ARNISON[j], A. ASTBURY[j], B. AUBERT[b], C. BACCI[i], G. BAUER[l], A. BÉZAGUET[d], R. BÖCK[d], T.J.V. BOWCOCK[f], M. CALVETTI[d], T. CARROLL[d], P. CATZ[b], P. CENNINI[d], S. CENTRO[d], F. CERADINI[d], S. CITTOLIN[d], D. CLINE[l], C. COCHET[k], J. COLAS[b], M. CORDEN[c], D. DALLMAN[d], M. DeBEER[k], M. DELLA NEGRA[b], M. DEMOULIN[d], D. DENEGRI[k], A. Di CIACCIO[i], D. DiBITONTO[d], L. DOBRZYNSKI[g], J.D. DOWELL[c], M. EDWARDS[c], K. EGGERT[a], E. EISENHANDLER[f], N. ELLIS[d], P. ERHARD[a], H. FAISSNER[a], G. FONTAINE[g], R. FREY[h], R. FRÜHWIRTH[l], J. GARVEY[c], S. GEER[g], C. GHESQUIÈRE[g], P. GHEZ[b], K.L. GIBONI[a], W.R. GIBSON[f], Y. GIRAUD-HÉRAUD[g], A. GIVERNAUD[k], A. GONIDEC[b], G. GRAYER[j], P. GUTIERREZ[h], T. HANSL-KOZANECKA[a], W.J. HAYNES[j], L.O. HERTZBERGER[2], C. HODGES[h], D. HOFFMANN[a], H. HOFFMANN[d], D.J. HOLTHUIZEN[2], R.J. HOMER[c], A. HONMA[f], W. JANK[d], G. JORAT[d], P.I.P. KALMUS[f], V. KARIMÄKI[e], R. KEELER[f], I. KENYON[c], A. KERNAN[h], R. KINNUNEN[e], H. KOWALSKI[d], W. KOZANECKI[h], D. KRYN[d], F. LACAVA[d], J.-P. LAUGIER[k], J.-P. LEES[b], H. LEHMANN[a], K. LEUCHS[a], A. LÉVÊQUE[k], D. LINGLIN[b], E. LOCCI[k], M. LORET[k], J.-J. MALOSSE[k], T. MARKIEWICZ[d], G. MAURIN[d], T. McMAHON[c], J.-P. MENDIBURU[g], M.-N. MINARD[b], M. MORICCA[i], H. MUIRHEAD[d], F. MULLER[d], A.K. NANDI[j], L. NAUMANN[d], A. NORTON[d], A. ORKIN-LECOURTOIS[g], L. PAOLUZI[i], G. PETRUCCI[d], G. PIANO MORTARI[i], M. PIMIÄ[e], A. PLACCI[d], E. RADERMACHER[a], J. RANSDELL[h], H. REITHLER[a], J.-P. REVOL[d], J. RICH[k], M. RIJSSENBEEK[d], C. ROBERTS[j], J. ROHLF[d], P. ROSSI[d], [redacted], B. SADOULET[d], G. SAJOT[g], G. SALVI[f], G. SALVINI[i], J. SASS[k], J. SAUDRAIX[k], A. SAVOY-NAVARRO[k], D. SCHINZEL[f], W. SCOTT[j], T.P. SHAH[j], M. SPIRO[k], J. STRAUSS[l], K. SUMOROK[c], F. SZONCSO[l], D. SMITH[h], C. TAO[d], G. THOMPSON[f], J. TIMMER[d], E. TSCHESLOG[a], J. TUOMINIEMI[e], [redacted], J.-P. VIALLE[d], J. VRANA[g], V. VUILLEMIN[d], H.D. WAHL[l], P. WATKINS[c], J. WILSON[c], Y.G. XIE[d], M. YVERT[b] and E. ZURFLUH[d]

Aachen[a]–Annecy (LAPP)[b]–Birmingham[c]–CERN[d]–Helsinki[e]–Queen Mary College, London[f]–Paris (Coll. de France)[g]–Riverside[h]–Rome[i]–Rutherford Appleton Lab.[j]–Saclay (CEN)[k]–Vienna[l] Collaboration

Received 23 January 1983

Shen & Barabási, PNAS, 2014

Credit allocation

Problem:

How to allocate credit for multi-author publications?

Challenge:

1. Multiple authorship breaks the symmetry between contribution and credit.
2. It is hard to quantify the actual contribution of authors, especially for those outside of the particular research field.
3. Each discipline runs its own informal credit allocation system.

Shen & Barabási, PNAS, 2014

Credit allocation

Existing methods:

- View each author of a multi-author publication as the sole author [Garfield, Science, 1972]
  - Causing inflated scientific impact for publications with multiple authors.
- Allocate fractional credit evenly among coauthors, assuming they contribute equally to a publication [Hirsch, PNAS, 2005]
  - Failing to account for the fact that authors' contributions are never equal, hence dilates the credit of the intellectual leader in a discovery.
- Allocate credit according to the order or role of coauthors. [Hagen, PLoS ONE, 2008][Stallings, PNAS, 2013]
  - The agreed-upon rules for author list vary from discipline to discipline.
  - For example:
    - In computer science, the rank of authors reflects a decreasing degree of contribution;
    - In biology, the first and last authors get the lion's share of credit;
    - In most physical sciences, the corresponding author gets the most credit;

We lack a discipline-independent method to decipher the informal credit allocation process in science.

Credit allocation

Case study:

Case A

2010 Nobel Prize in Chemistry

Baba, Negishi, J. Am. Chem. Soc. 98, 6729 (1976) 

Case B

2010 Nobel Prize in Physics

Novoselov, Geim, Science, 306, 666 (2004)

Frequently co-cited papers:

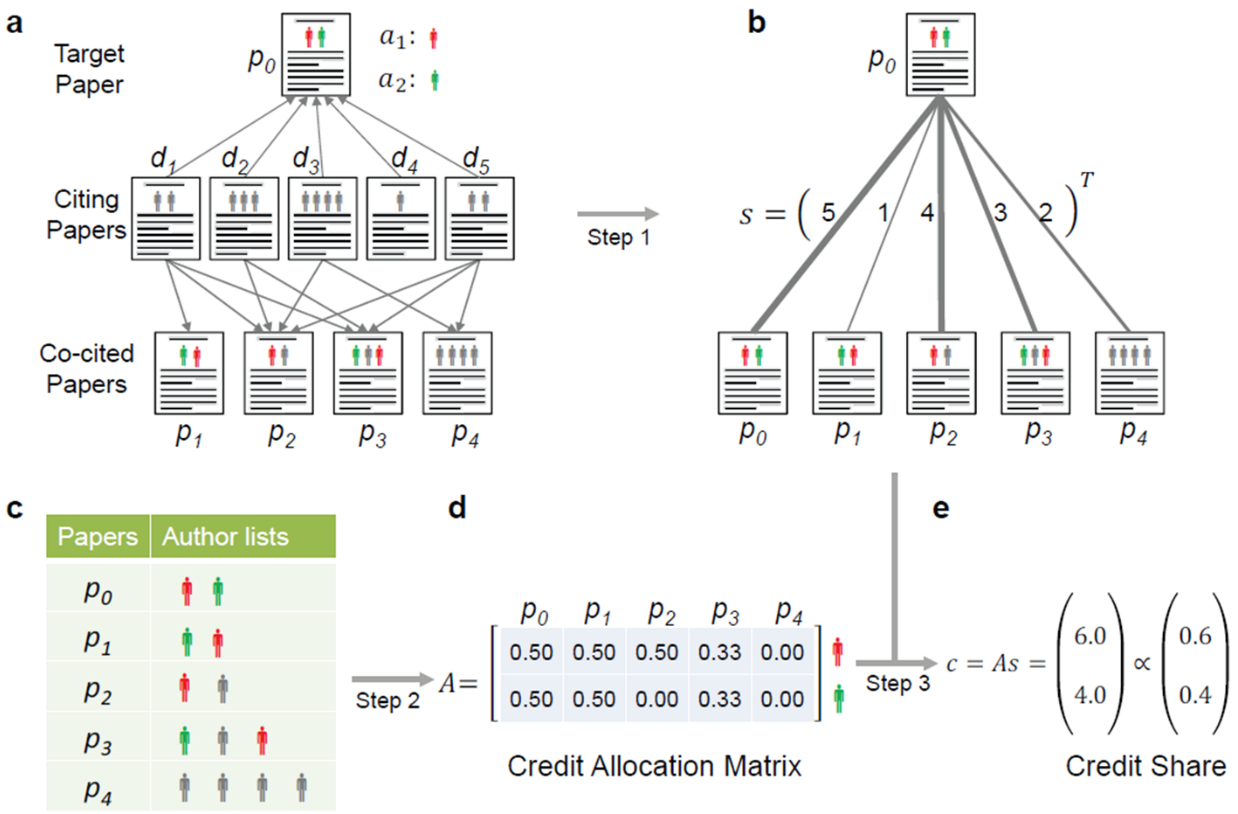
1. Negishi, Okukado, King,Van Horn, Spiegel, J. Am. Chem. Soc. (1978)
2. Negishi, King, Okukado, J. Org. Chem. (1977)
3. Negishi, Vanhorn, J. Am. Chem. Soc. (1977)
4. Negishi, Vanhorn, J. Am. Chem. Soc. (1978)
5. Negishi, Valente. Kobayashi, J. Am. Chem. Soc. (1980)

Frequently co-cited papers:

1. Geim, Novoselov, Nature (2007)
2. Novoselov, Jiang, Schedin, Booth, Khotkevich, Morozov, Geim, PNAS (2005)
3. Novoselov, Geim, Morozov, Jiang, Katsnelson, rigorieva, Dubonos, Firsov, Nature (2005)
4. Castro Neto, Guinea, Peres, Novoselov, Geim, Rev. Mod. Phys. (2009)
5. Ferrari, Meyer, Scardaci, Casiraghi, Lazzeri, auri,Piscanec. Jiang, Novoselov, Roth, Geim. Phys. Rev. Lett. (2006)

Shen & Barabási, PNAS, 2014

Credit allocation



Co-cited papers:

Co-citation strength *s*

Credit allocation matrix *A*

Credit share:

c=As

Shen & Barabási, PNAS, 2014

Credit allocation

Case revisiting :

Case A

2010 Nobel Prize in Chemistry

Baba, Negishi, J. Am. Chem. Soc. 98, 6729 (1976) 

Frequently co-cited papers:

1. **Negishi, Okukado, King,Van Horn, Spiegel, J. Am. Chem. Soc. (1978)**
2. **Negishi, King, Okukado, J. Org. Chem. (1977)**
3. **Negishi, Vanhorn, J. Am. Chem. Soc. (1977)**
4. **Negishi, Vanhorn, J. Am. Chem. Soc. (1978)**
5. **Negishi, Valente. Kobayashi, J. Am. Chem. Soc. (1980)**

Credit share: (0.28, 0.72)

Case B

2010 Nobel Prize in Physics

Novoselov, Geim, Science, 306, 666 (2004)

Frequently co-cited papers:

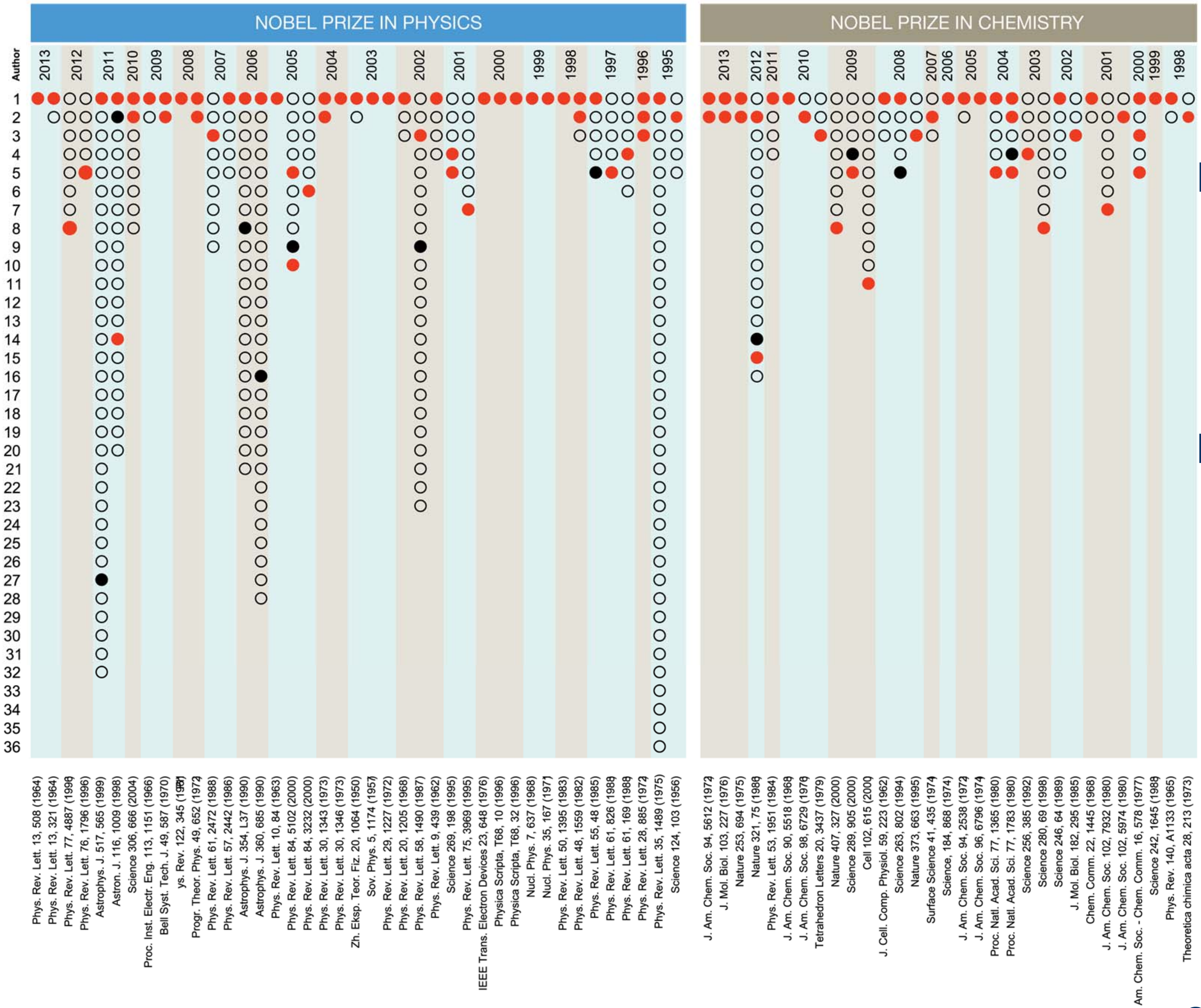
1. **Geim, Novoselov, Nature (2007)**
2. **Novoselov, Jiang, Schedin, Booth, Khotkevich, Morozov, Geim, PNAS (2005)**
3. **Novoselov, Geim, Morozov, Jiang, Katsnelson, rigorieva, Dubonos, Firsov, Nature (2005)**
4. **Castro Neto, Guinea, Peres, Novoselov, Geim, Rev. Mod. Phys. (2009)**
5. **Ferrari, Meyer, Scardaci, Casiraghi, Lazzeri, auri,Piscanec. Jiang, Novoselov, Roth, Geim. Phys. Rev. Lett. (2006)**

Credit share: (0.5, 0.5)

Shen & Barabási, PNAS, 2014

Credit allocation

Validation



Datasets:

APS: American Physical Society
WOS: Web of science
Nobel prize-winning papers

Metric:

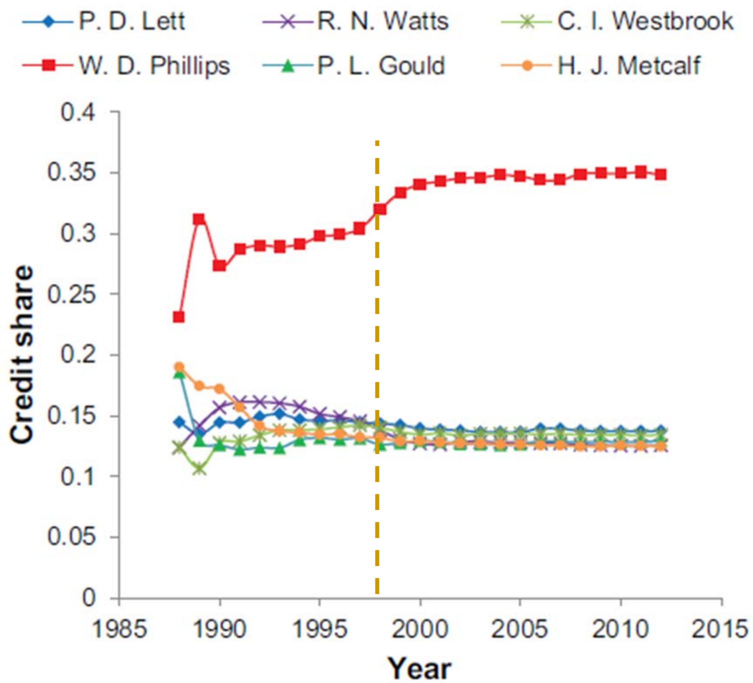
Whether our method could identify the Nobel Laureates from the author list.

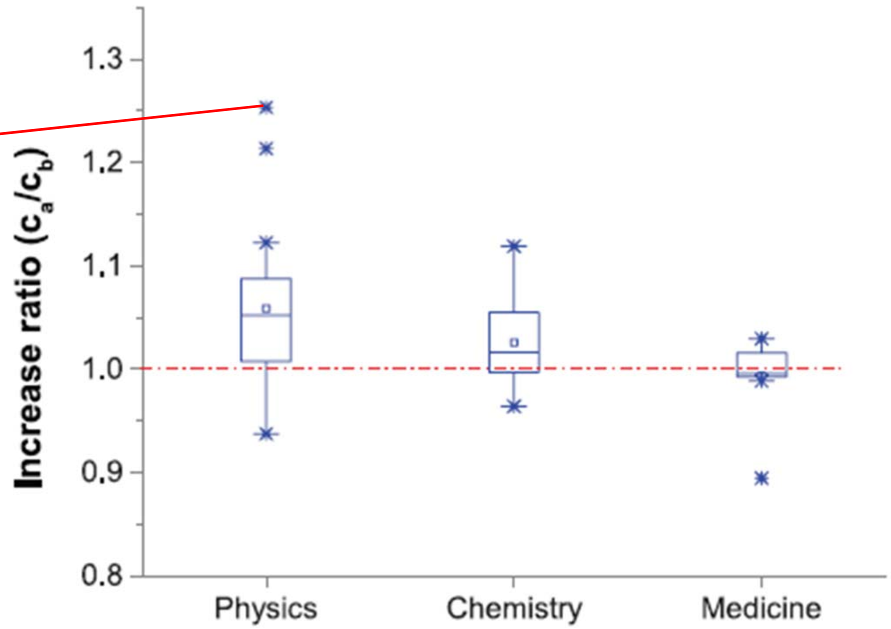
Correct at 51 of 63 test cases.

Shen & Barabási, PNAS, 2014

Credit allocation

Credit share evolution





c_a: average credit share over 3 years after publication;

c_b: average credit share over 3 years before publication;

Increase ratio: c_a / c_b

Shen & Barabási, PNAS, 2014

Credit allocation

Comparing independent authors

Three independent papers (six scientists) contribute to the discovery of Higgs Boson.

VOLUME 13, NUMBER 9 PHYSICAL REVIEW LETTERS 31 AUGUST 1964

BROKEN SYMMETRY AND THE MASS OF GAUGE VECTOR MESONS*

F. Englert and R. Brout
Faculté des Sciences, Université Libre de Bruxelles, Bruxelles, Belgium
(Received 26 June 1964)

VOLUME 13, NUMBER 16 PHYSICAL REVIEW LETTERS 19 OCTOBER 1964

BROKEN SYMMETRIES AND THE MASSES OF GAUGE BOSONS

Peter W. Higgs
Tait Institute of Mathematical Physics, University of Edinburgh, Edinburgh, Scotland
(Received 31 August 1964)

VOLUME 13, NUMBER 20 PHYSICAL REVIEW LETTERS 16 NOVEMBER 1964

GLOBAL CONSERVATION LAWS AND MASSLESS PARTICLES*

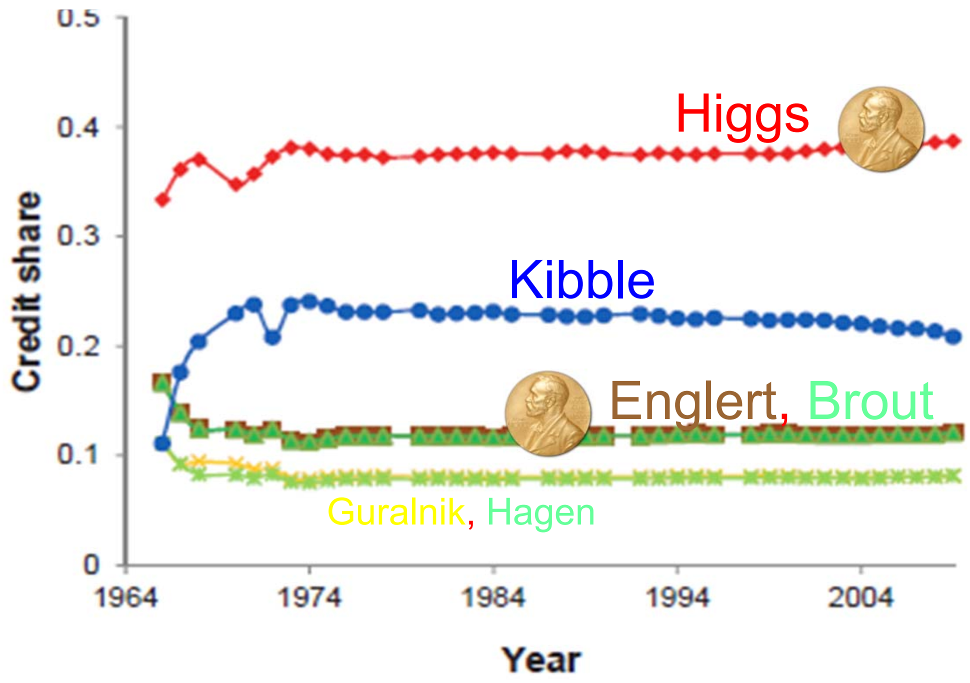
G. S. Guralnik,† C. R. Hagen,‡ and T. W. B. Kibble
Department of Physics, Imperial College, London, England
(Received 12 October 1964)

Who gets the Nobel prize, i.e., who gets high credit from the Nobel committee?

Shen & Barabási, PNAS, 2014

Credit allocation

Comparing independent authors



Higgs & Englert

Kibble

"I really rather hoped before the announcement that they would make the number up to three, and <u>there was certainly an obvious candidate to be the third, Tom Kibble</u>"

(Peter Higgs, BBC Interview 2014)

Credit allocation

Summary

- We developed a method to quantify the credit share of coauthors by reproducing the collective credit allocation process informally used by the scientific community.
  - Credit is allocated among coauthors based on their perceived contribution rather than their actual contribution;
  - Established scientists receive more credit than their junior collaborators from their coauthored publication
    - This situation can change, however, if the junior one makes important independent contribution to the field
  - Credit share is a dynamic quantify the changes with the evolution of the field

THANK YOU !