Improving Ranking Consistency for Web Search by Leveraging a Knowledge Base and Search Logs

Jyun-Yu Jiang[†], Jing Liu[‡], Chin-Yew Lin[‡] and Pu-Jen Cheng[†]

[†]National Taiwan University
[‡]Microsoft Research

October 22, 2015 (CIKM)

Outline for Section 1

1. Introduction
   - Relevance Ranking in Web Search
   - Ranking Consistency
2. Ranking Consistency in Web Search
3. Consistent Ranking Model (Stage 1)
4. Ensemble-based Re-ranking (Stage 2)
5. Experiments
6. Conclusions and Future Work

Why Relevance Ranking in Web Search?

- Sort web pages by users' information needs
  - The higher the rank, the more relevant the page.
- Filter out irrelevant web pages for users
  - More than 968 million websites in 2014

[http://www.internetlivestats.com/]

The Goal of Relevance Ranking in Web Search

Estimate the relevance of each web page to a query,
and then return a ranked list of web pages with higher relevance.

Conventional Approaches

Bag-of-Words Retrieval Model

- Retrieval Functions
  - TF-IDF, Okapi-BM25, etc.
- Adjust by User Feedback
  - e.g., Rocchio relevance feedback

Learning to Rank

- Feature Extraction
  - e.g., contents and URLs
- Supervised Ranking Model
  - RankNet, LambdaMART, etc.

Specialized Web Search

- Personalized Search
- Federated Web Search

Accurate Evaluation

- Rank of rel. docs (e.g., NDCG)
- User click-through data

However, all of previous work focus on optimizing queries separately.

An example from a commercial search engine

- There are search results of three basketball players in NBA.

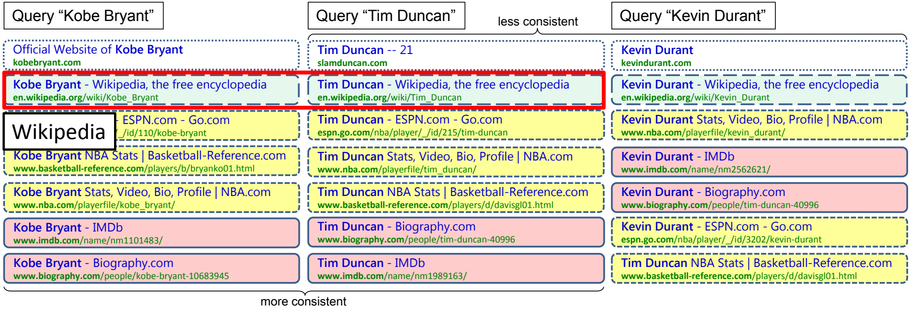
less consistent

Query "Kobe Bryant"	Query "Tim Duncan"	Query "Kevin Durant"
Official Website of **Kobe Bryant** kobebryant.com	**Tim Duncan** -- 21 slamduncan.com	**Kevin Durant** kevindurant.com
Kobe Bryant - Wikipedia, the free encyclopedia en.wikipedia.org/wiki/Kobe_Bryant	**Tim Duncan** - Wikipedia, the free encyclopedia en.wikipedia.org/wiki/Tim_Duncan	**Kevin Durant** - Wikipedia, the free encyclopedia en.wikipedia.org/wiki/Kevin_Durant
Kobe Bryant Stats - ESPN.com - Go.com espn.go.com/nba/player/_/id/110/kobe-bryant	**Tim Duncan** - ESPN.com - Go.com espn.go.com/nba/player/_/id/215/tim-duncan	**Kevin Durant** Stats, Video, Bio, Profile | NBA.com www.nba.com/playerfile/kevin_durant/
Kobe Bryant NBA Stats | Basketball-Reference.com www.basketball-reference.com/players/b/bryanko01.html	**Tim Duncan** Stats, Video, Bio, Profile | NBA.com www.nba.com/playerfile/tim_duncan/	**Kevin Durant** - IMDb www.imdb.com/name/nm2562621/
Kobe Bryant Stats, Video, Bio, Profile | NBA.com www.nba.com/playerfile/kobe_bryant/	**Tim Duncan** NBA Stats | Basketball-Reference.com www.basketball-reference.com/players/d/davisgl01.html	**Kevin Durant** - Biography.com www.biography.com/people/tim-duncan-40996
Kobe Bryant - IMDb www.imdb.com/name/nm1101483/	**Tim Duncan** - Biography.com www.biography.com/people/tim-duncan-40996	**Kevin Durant** - ESPN.com - Go.com espn.go.com/nba/player/_/id/3202/kevin-durant
Kobe Bryant - Biography.com www.biography.com/people/kobe-bryant-10683945	**Tim Duncan** - IMDb www.imdb.com/name/nm1989163/	**Tim Duncan** NBA Stats | Basketball-Reference.com www.basketball-reference.com/players/d/davisgl01.html

more consistent

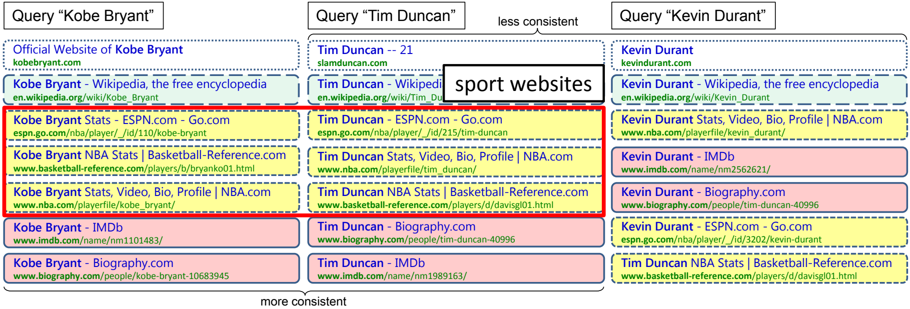
An example from a commercial search engine

- There are search results of three basketball players in NBA.



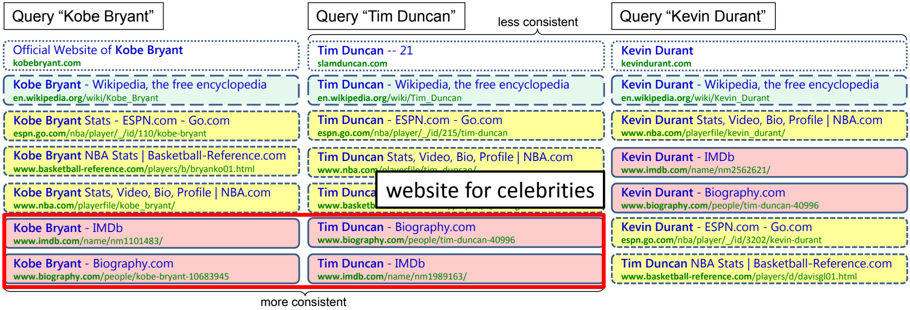
An example from a commercial search engine

- There are search results of three basketball players in NBA.



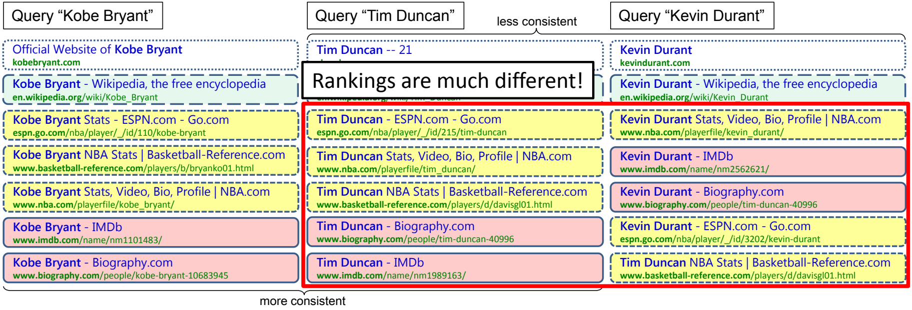
An example from a commercial search engine

- There are search results of three basketball players in NBA.



An example from a commercial search engine

- There are search results of three basketball players in NBA.



Outline for Section 2

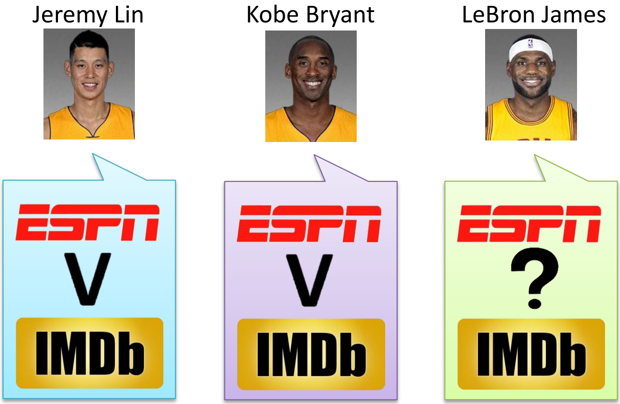
Topical Clusters in Websites

- Web pages about a topic on a website can be treated a topical cluster

Can these information be well exploited?

While ranking web pages for three queries...

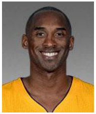
While ranking web pages for three queries...



While ranking web pages for three queries...

While ranking web pages for three queries...

We are all basketball players in NBA!

Jeremy Lin	Kobe Bryant	LeBron James
		
ESPN V IMDb	ESPN V IMDb	ESPN V IMDb

Ranking Consistency in Web Search

- The relevance of web pages in the same topical clusters would be consistent for similar search intents.
- Ranking consistency may help the relevance ranking in web search.

The goal of this paper

Learn the ranking consistency, and then improve the relevance ranking

Do users realize the ranking consistency?

- Two user surveys via Amazon Mechanical Turk (MTurk)
  - a crowdsourcing platform for work that requires human intelligence

amazon mechanical turk™
Artificial Artificial Intelligence

- Five pairs of queries of five different types
- Re-rank results to be consistent, and then ask whether are improved

First Survey

- observe two results together
- 25 questions for 10 workers

Second Survey

- observe two results separately
- 50 questions for 10 workers

User Survey Results

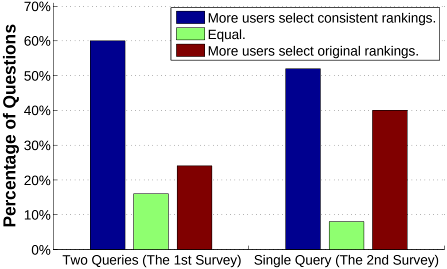


Figure: The results of two user surveys.

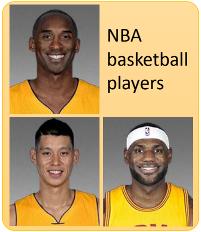
- consistent $\gg$ original in 1st survey
  - consistency can be observed **directly**
- consistent $>$ original in 2nd survey
  - even users **cannot** observe directly!

The ranking consistency is **realizable** and **more preferable**!

Two Challenges in Ranking Consistency

Challenge 1

- How to determine **similar-intent queries** and **topical clusters**?
- How to consistently rank **topical clusters**?



similar-intent queries

>

topical clusters

Two Challenges in Ranking Consistency (Cont'd)

Challenge 2

- How to handle web pages not in any topical cluster?
  - e.g., **official websites** and **personal web pages**

Jeremy Lin

JLIN7: Lin's official website

Moreover, the best ranking for each query might be a little different.

Two-stage Re-ranking Model

Stage 1: Consistent Ranking Model

- Apply a **knowledge base** to determine queries with similar intents
- Establish topical clusters by **URL patterns**
- Learn the relevance of topical clusters from **click-through logs**

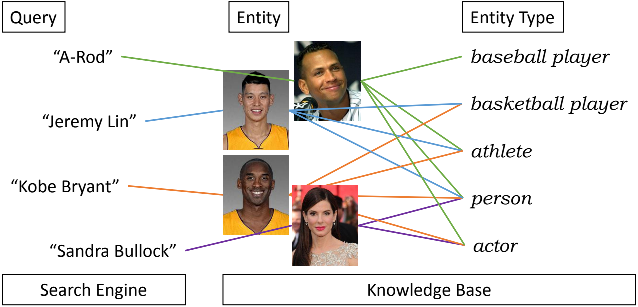
Stage 2: Ensemble-based Re-ranking

- **Re-rank search results** of an original ranker with results from Stage 1
- Apply **several features** to decide the parameter in the ensemble

Outline for Section 3

Query Intents and Entity Types

- Named entity recognition in queries (NERQ)
  - More than 70% queries cover entities [Guo *et al.*, SIGIR'09].
- Entity types in knowledge bases
  - Knowledge bases summarize **entity types**.
  - Queries with **same types** may share **similar intents**



Query Intents and Entity Types

- Named entity recognition in queries (NERQ)
  - More than 70% queries cover entities [Guo *et al.*, SIGIR'09].
- Entity types in knowledge bases
  - Knowledge bases summarize **entity types**.
  - Queries with **same types** may share **similar intents**

Query

Entity

Entity Type

"A-Rod"

baseball player

"Jeremy Lin"

"Kobe Bryant"

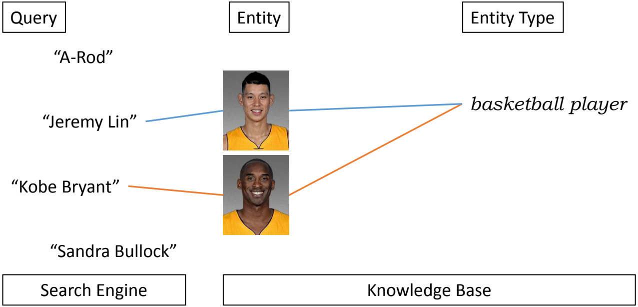
"Sandra Bullock"

Search Engine

Knowledge Base

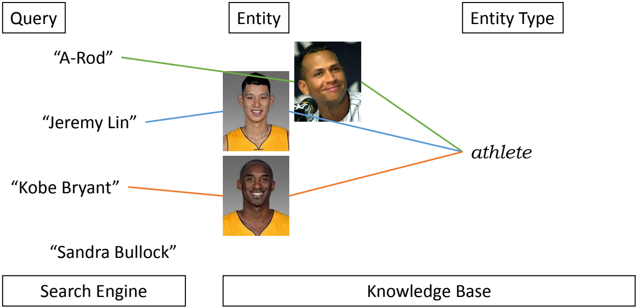
Query Intents and Entity Types

- Named entity recognition in queries (NERQ)
  - More than 70% queries cover entities [Guo *et al.*, SIGIR'09].
- Entity types in knowledge bases
  - Knowledge bases summarize **entity types**.
  - Queries with **same types** may share **similar intents**



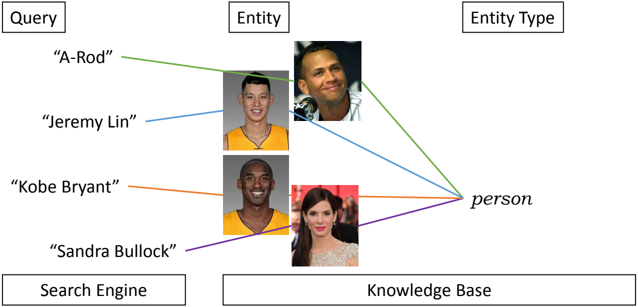
Query Intents and Entity Types

- Named entity recognition in queries (NERQ)
  - More than 70% queries cover entities [Guo *et al.*, SIGIR'09].
- Entity types in knowledge bases
  - Knowledge bases summarize **entity types**.
  - Queries with **same types** may share **similar intents**



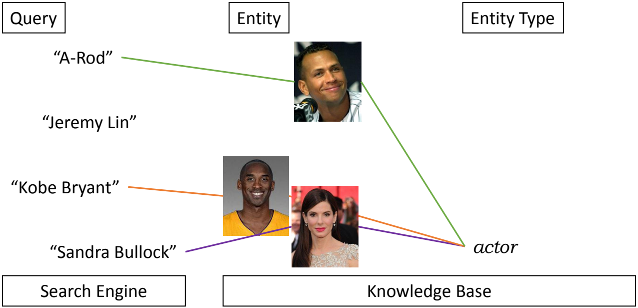
Query Intents and Entity Types

- Named entity recognition in queries (NERQ)
  - More than 70% queries cover entities [Guo *et al.*, SIGIR'09].
- Entity types in knowledge bases
  - Knowledge bases summarize **entity types**.
  - Queries with **same types** may share **similar intents**



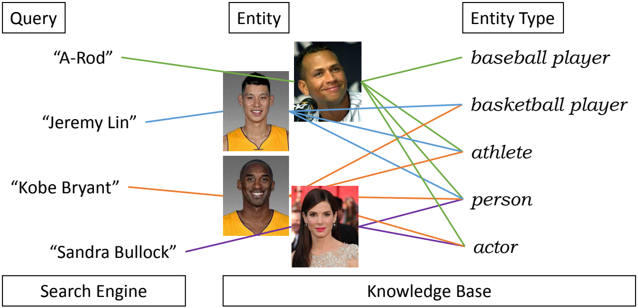
Query Intents and Entity Types

- Named entity recognition in queries (NERQ)
  - More than 70% queries cover entities [Guo *et al.*, SIGIR'09].
- Entity types in knowledge bases
  - Knowledge bases summarize **entity types**.
  - Queries with **same types** may share **similar intents**



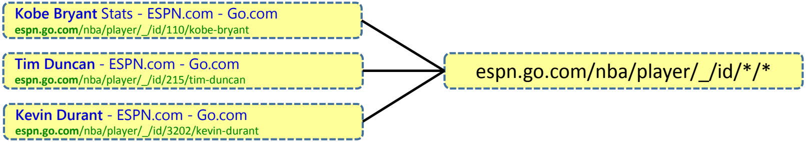
Query Intents and Entity Types

- Named entity recognition in queries (NERQ)
  - More than 70% queries cover entities [Guo *et al.*, SIGIR'09].
- Entity types in knowledge bases
  - Knowledge bases summarize **entity types**.
  - Queries with **same types** may share **similar intents**



Topical Clusters and URL Patterns

- Some websites have specific contents.
    - e.g., `ESPN.com` contains web pages about different sports.
- Pages in the same topical cluster usually share the same URL pattern.
    - `espn.go.com/nba/player/_/id/*/*` is for basketball players.
    - `espn.go.com/mlb/player/_/id/*/*` is for baseball players.
- URL Pattern Extraction [Jiang *et al.*, WWW'12]
    - Collect URL collection from search logs
    - Generalize URLs into regular expressions as URL patterns



Consistent Ranking Model (Stage 1)

Assumption 1

- The relevance of a page is decided by the pattern relevance.

$$P(u \mid q) = \begin{cases} P(p \mid q) & \text{, if } u \text{ is matched by pattern } p \\ 0 & \text{, otherwise} \end{cases}$$

Assumption 2

- The relevance distribution is an aggregation over query types.

$$P(p \mid q) = \sum_{t \in T(q)} P(p \mid t, q) \cdot P(t \mid q) = \sum_{t \in T(q)} P(p \mid t) \cdot P(t \mid q)$$

Question: How to compute $P(p \mid t)$ and $P(t \mid q)$?

Pattern-Type Relevance and Type Distribution

Pattern-Type Relevance $P(p \mid t)$

- Extract pairwise preference (p_1, p_2) from user feedback in logs
- Aggregate (p_1, p_2) into (p_1, p_2, w) by each entity to avoid popularity biases
- Learn the relevance $P(p \mid t)$ by rank aggregation

Type Distribution $P(t \mid q)$

- Estimate how much users treat q is a query of the type t.
- Adopt click-through data and Bayesian m-estimate smoothing

$$P(t \mid q) = \frac{\sum_{p \in S} P(t \mid p) \cdot Click(p, q) + m \cdot P(t)}{m + \sum_{t \in T(q)} \sum_{p \in S} P(t \mid p) \cdot Click(p, q)}$$

See our paper for detailed information.

Outline for Section 4

Ensemble-based Re-ranking (Stage 2)

- Ensemble of the consistent ranking model and the original ranker
- For a query q and a URL u ranked in the position i by the original ranker, the relevance can be computed as:

$$P(u \mid q, i) = \underbrace{\lambda \cdot P(u \mid q)}_{\text{Consistency Ranking}} + \underbrace{(1-\lambda) \cdot P(u \mid i)}_{\text{Original Ranker}},$$

Multiple Parameters

- In different cases, the best parameter might be also different.
  - e.g., λ should be lower for pages in personal sites.
- Replace the parameter λ with a logistic function
  - Then some features can be the input and adjust the parameter!

$$\lambda(X) = \frac{1}{1 + \exp(-f(X))}, \quad f(X) = \beta_0 + \sum_{i=1}^{|X|} \beta_i \cdot x_i$$

  - x_i is the i-th feature in the feature set X
  - β_0 is the bias parameter
- $|X| = 0$ is a special case of single parameter.
- Use the RankNet cost function to optimize β parameters.

See our paper for detailed optimization.

Re-ranking Features

- Various features in three levels are considered.
    - including query features, entity features and URL features
- Helpful in recognizing different situations and adjusting the parameter

Query Features

- query length
- query frequency

Entity Features

- number of types
- type entropy
- entity frequency

URL Features

- pattern matching
- original position
- consistent relevance
- N-gram similarity
    - host and URL

There are totally 10 features used in our approach.

Outline for Section 5

Dataset and Experimental Settings

Knowledge Base

- Freebase dumped in January 2014
- Remove very spare types with less than 5 entities
- Finally 444 types in the type set

Search Engine Logs

- Logs of a commercial search engine in November 2013
- 56,466,534 queries for 847,682 distinct entities after extraction
- Queries of 21 days as training data, the remaining as testing data
- Treat URLs with SAT-Clicks as the ground-truth

Dataset and Experimental Settings (Cont'd)

- Seven testing subsets with different conditions

Dataset	Description
All	All queries in testing data.
Head	Queries with top 10% entity frequency.
Tail	Queries with bottom 10% entity frequency.
New	Queries which do not appear in training data.
Peo.	Queries with type `people/person`.
Loc.	Queries with type `location/location`.
Org.	Queries with type `organization/organization`.

`person`, `location` and `organization` cover **most entities**

Evaluation of Ranking Consistency

- Propose a new metric based on Kendall's tau
- For a type t and a set of queries $Q(t)$ with the type t

$$\frac{1}{\binom{|Q(t)|}{2}} \sum_{q_1 \in Q(t)} \sum_{q_2 \in Q(t) \setminus q_1} \tau\left(r\left(q_1, t\right), r\left(q_2, t\right)\right)$$

  - $r(q, t)$ denotes the ranking result of t's URL patterns with query q.
  - $\tau(r_1, r_2)$ is the standard Kendall's tau rank correlation coefficient.
- Give zero rank scores to patterns without appearance in search results

Baseline

- The original ranker in the search engine (Default).

Evaluation of Ranking Consistency (Cont'd)

Type	Default	Our Approach
Overall types	0.5671	0.5943 (+4.78%)
people/person	0.6410	0.6517 (+1.67%)
location/location	0.6327	0.6455 (+2.02%)
organization/organization	0.7533	0.7588 (+0.73%)
celebrities/celebrity	0.6306	0.6697 (+6.21%)
music/album	0.4589	0.4842 (+5.51%)
book/book	0.5367	0.5544 (+3.31%)

Our method significantly improved the ranking consistency.

Evaluation of Ranking Consistency (Cont'd)

Type	Default	Our Approach
Overall types	0.5671	0.5943 (+4.78%)
people/person	0.6410	0.6517 (+1.67%)
location/location	0.6327	0.6455 (+2.02%)
organization/organization	0.7533	0.7588 (+0.73%)
celebrities/celebrity	0.6306	0.6697 (+6.21%)
music/album	0.4589	0.4842 (+5.51%)
book/book	0.5367	0.5544 (+3.31%)

people and location improve less than overall because **they are too general**.

Evaluation of Ranking Consistency (Cont'd)

Type	Default	Our Approach
Overall types	0.5671	0.5943 (+4.78%)
people/person	0.6410	0.6517 (+1.67%)
location/location	0.6327	0.6455 (+2.02%)
organization/organization	0.7533	0.7588 (+0.73%)
celebrities/celebrity	0.6306	0.6697 (+6.21%)
music/album	0.4589	0.4842 (+5.51%)
book/book	0.5367	0.5544 (+3.31%)

celebrity improves the most because **many sites are about celebrities**.

Evaluation of Ranking Consistency (Cont'd)

Type	Default	Our Approach
Overall types	0.5671	0.5943 (+4.78%)
people/person	0.6410	0.6517 (+1.67%)
location/location	0.6327	0.6455 (+2.02%)
organization/organization	0.7533	0.7588 (+0.73%)
celebrities/celebrity	0.6306	0.6697 (+6.21%)
music/album	0.4589	0.4842 (+5.51%)
book/book	0.5367	0.5544 (+3.31%)

organization improves the least because they usually have **only official sites**.

Evaluation of Re-ranking Models

Baseline Method

- the default ranker of that commercial search engine
  - a learning-to-rank model with myriad features
- a strong competitor (as a product in the **real-world search engine**)
- evaluate **whether the re-ranking model is effective**

Evaluation Measure

- Mean Average Precision (MAP)
  - consider all relevant (clicked) documents
- Mean Reciprocal Rank (MRR)
  - consider the first predicted relevant document

Evaluation of Re-ranking Models (Cont'd)

		Default	Our Approach (single params)	Our Approach (multiple params)
All	MAP	0.7272	0.7454 (+2.49%)	**0.7571 (+4.12%)**
	MRR	0.7288	0.7469 (+2.49%)	**0.7589 (+4.13%)**
Head	MAP	0.7294	0.7491 (+2.70%)	**0.7611 (+4.34%)**
	MRR	0.7309	0.7505 (+2.68%)	**0.7627 (+4.35%)**
Tail	MAP	0.7116	0.7228 (+1.57%)	**0.7384 (+3.76%)**
	MRR	0.7138	0.7251 (+1.58%)	**0.7408 (+3.78%)**
New	MAP	0.7272	0.7453 (+2.49%)	**0.7572 (+3.83%)**
	MRR	0.7287	0.7468 (+2.48%)	**0.7589 (+3.83%)**
Peo.	MAP	0.7468	0.7756 (+3.86%)	**0.7834 (+4.89%)**
	MRR	0.7483	0.7772 (+3.86%)	**0.7851 (+4.92%)**
Loc.	MAP	0.7268	0.7465 (+2.72%)	**0.7573 (+4.20%)**
	MRR	0.7283	0.7481 (+2.71%)	**0.7588 (+4.19%)**
Org.	MAP	0.8422	0.8615 (+2.28%)	**0.8674 (+2.99%)**
	MRR	0.8432	0.8624 (+2.28%)	**0.8684 (+3.00%)**

Evaluation of Re-ranking Models (Cont'd)

		Default	Our Approach (single params)	Our Approach (multiple params)
All	MAP	0.7272	0.7454 (+2.49%)	**0.7571 (+4.12%)**
	MRR	0.7288	0.7469 (+2.49%)	**0.7589 (+4.13%)**
Head	MAP	0.7294	0.7491 (+2.70%)	**0.7611 (+4.34%)**
	MRR	0.7309	0.7505 (+2.68%)	**0.7627 (+4.35%)**
Tail	MAP	0.7116	0.7228 (+1.57%)	**0.7384 (+3.76%)**
	MRR	0.7138	0.7251 (+1.58%)	**0.7408 (+3.78%)**

For the default ranking, head/tail queries are better/lower performance.

Evaluation of Re-ranking Models (Cont'd)

		Default	Our Approach (single params)	Our Approach (multiple params)
All	MAP	0.7272	0.7454 (+2.49%)	**0.7571 (+4.12%)**
	MRR	0.7288	0.7469 (+2.49%)	**0.7589 (+4.13%)**
Peo.	MAP	0.7468	0.7756 (+3.86%)	**0.7834 (+4.89%)**
	MRR	0.7483	0.7772 (+3.86%)	**0.7851 (+4.92%)**
Loc.	MAP	0.7268	0.7465 (+2.72%)	**0.7573 (+4.20%)**
	MRR	0.7283	0.7481 (+2.71%)	**0.7588 (+4.19%)**
Org.	MAP	0.8422	0.8615 (+2.28%)	**0.8674 (+2.99%)**
	MRR	0.8432	0.8624 (+2.28%)	**0.8684 (+3.00%)**

Peo. and Org. have better performance because they have own official sites.

Evaluation of Re-ranking Models (Cont'd)

		Default	Our Approach (single params)	Our Approach (multiple params)
All	MAP	0.7272	0.7454 (+2.49%)	**0.7571 (+4.12%)**
	MRR	0.7288	0.7469 (+2.49%)	**0.7589 (+4.13%)**
Head	MAP	0.7294	0.7491 (+2.70%)	**0.7611 (+4.34%)**
	MRR	0.7309	0.7505 (+2.68%)	**0.7627 (+4.35%)**
Tail	MAP	0.7116	0.7228 (+1.57%)	**0.7384 (+3.76%)**
	MRR	0.7138	0.7251 (+1.58%)	**0.7408 (+3.78%)**
New	MAP	0.7272	0.7453 (+2.49%)	**0.7572 (+3.83%)**
	MRR	0.7287	0.7468 (+2.48%)	**0.7589 (+3.83%)**
Peo.	MAP	0.7468	0.7756 (+3.86%)	**0.7834 (+4.89%)**
	MRR	0.7483	0.7772 (+3.86%)	**0.7851 (+4.92%)**
Loc.	MAP	0.7268	0.7465 (+2.72%)	**0.7573 (+4.20%)**
	MRR	0.7283	0.7481 (+2.71%)	**0.7588 (+4.19%)**
Org.	MAP	0.8422	0.8615 (+2.28%)	**0.8674 (+2.99%)**
	MRR	0.8432	0.8624 (+2.28%)	**0.8684 (+3.00%)**

Using multiple parameters achieves better performance.

Evaluation of Re-ranking Models (Cont'd)

		Default	Our Approach (single params)	Our Approach (multiple params)
All	MAP	0.7272	0.7454 (+2.49%)	**0.7571 (+4.12%)**
	MRR	0.7288	0.7469 (+2.49%)	**0.7589 (+4.13%)**
Head	MAP	0.7294	0.7491 (+2.70%)	**0.7611 (+4.34%)**
	MRR	0.7309	0.7505 (+2.68%)	**0.7627 (+4.35%)**
Tail	MAP	0.7116	0.7228 (+1.57%)	**0.7384 (+3.76%)**
	MRR	0.7138	0.7251 (+1.58%)	**0.7408 (+3.78%)**

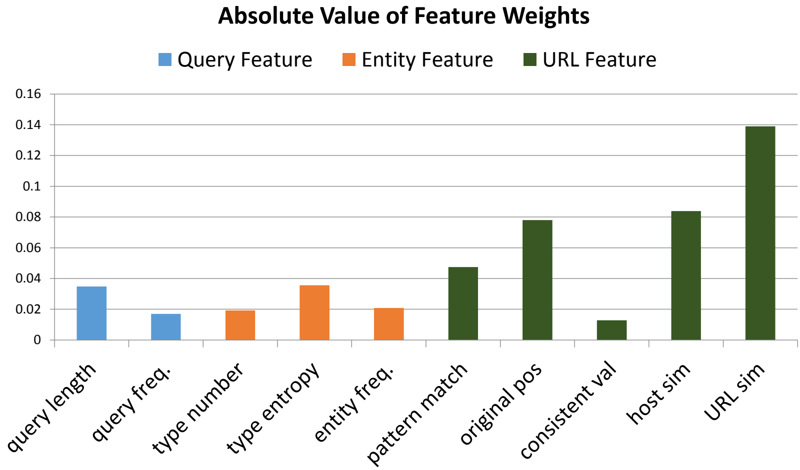
Although head queries are still better, tail queries have great improvements.

Evaluation of Re-ranking Models (Cont'd)

		Default	Our Approach (single params)	Our Approach (multiple params)
All	MAP	0.7272	0.7454 (+2.49%)	**0.7571 (+4.12%)**
	MRR	0.7288	0.7469 (+2.49%)	**0.7589 (+4.13%)**
Peo.	MAP	0.7468	0.7756 (+3.86%)	**0.7834 (+4.89%)**
	MRR	0.7483	0.7772 (+3.86%)	**0.7851 (+4.92%)**
Loc.	MAP	0.7268	0.7465 (+2.72%)	**0.7573 (+4.20%)**
	MRR	0.7283	0.7481 (+2.71%)	**0.7588 (+4.19%)**
Org.	MAP	0.8422	0.8615 (+2.28%)	**0.8674 (+2.99%)**
	MRR	0.8432	0.8624 (+2.28%)	**0.8684 (+3.00%)**

Peo. and Loc. improve more because they have many pages in topical clusters.
Org. improves less because they have few pages in topical clusters.

Feature Analysis



Outline for Section 6

Conclusions and Future Work

- Propose the new idea called ranking consistency in web search
- Two convincing user surveys on Amazon Mechanical Turk
- Propose a two-stage re-ranking model by leveraging a knowledge base and click-through data
- Propose features in three levels to adjust the ensemble
- Future Work
  - Supervised Approach
    - Optimize the ranking consistency and cost function at the same time
  - Unsupervised Approach
    - Consider the ranking consistency while calculating retrieval functions

Q & A

Thank you for listening! Any question?

Contact Information

- Jyun-Yu (John) Jiang
- http://jyunyu.csie.org/
- jyunyu.jiang@gmail.com

Illustration of Surveys in MTurk

Question 3

Upper 1(**jason maxiell**)

1: http://en.wikipedia.org/wiki/Jason_Maxiell
2: http://espn.go.com/nba/player/_/id/2775/jason-maxiell
3: http://www.rotoworld.com/player/nba/1155/jason-maxiell
4: http://www.basketball-reference.com/players/m/maxieja01.html
5: http://www.cbssports.com/nba/players/playerpage/555965/jason-maxiell

Upper 2(**mehmet okur**)

1: http://en.wikipedia.org/wiki/Mehmet_Okur
2: http://espn.go.com/nba/player/_/id/1014/mehmet-okur
3: http://www.basketball-reference.com/players/o/okurme01.html
4: http://www.rotoworld.com/player/nba/800/mehmet-okur
5: http://www.cbssports.com/nba/players/playerpage/240303/mehmet-okur

○ Upper rankings(↑) are more relevant.

◉ Equal.

○ Lower rankings(↓) are more relevant.

Lower 1(**jason maxiell**)

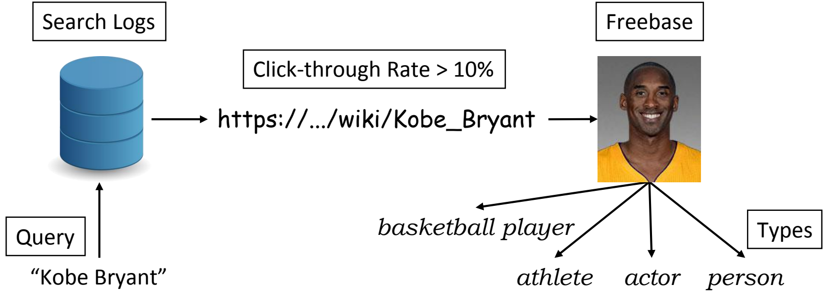
1: http://en.wikipedia.org/wiki/Jason_Maxiell
2: http://espn.go.com/nba/player/_/id/2775/jason-maxiell
3: http://www.basketball-reference.com/players/m/maxieja01.html
4: http://www.rotoworld.com/player/nba/1155/jason-maxiell
5: http://www.cbssports.com/nba/players/playerpage/555965/jason-maxiell

Lower 2(**mehmet okur**)

1: http://en.wikipedia.org/wiki/Mehmet_Okur
2: http://espn.go.com/nba/player/_/id/1014/mehmet-okur
3: http://www.basketball-reference.com/players/o/okurme01.html
4: http://www.rotoworld.com/player/nba/800/mehmet-okur
5: http://www.cbssports.com/nba/players/playerpage/240303/mehmet-okur

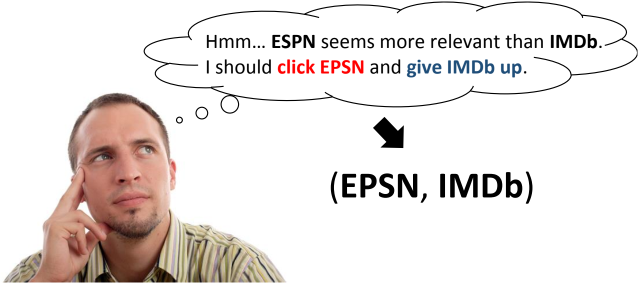
Query Type Extraction

- Any method of NERQ can be applied.
- Here the **click-through data** and **Wikipedia** is simply exploited.



Pattern-Type Relevance $P(p \mid t)$

- Exploit **click-through data** from search engine logs
  - Extract **pairwise preference** from original rankings and user feedback
- (p_1, p_2) denotes that the pattern p_1 is more relevant than p_2.



(EPSN, IMDb)

Preference Extraction

- A user searches "Kobe Bryant," and then clicks some web pages.

1. Kobe Bryant - Wikipedia, the free encyclopedia
   en.wikipedia.org/wiki/Kobe_Bryant
   $p_1 = \texttt{en.wikipedia.org/wiki/*}$
2. **(Clicked) KB24 - Official Website of Kobe Bryant**
   kobebryant.com, $p_2 = \emptyset$
3. **(Clicked) Kobe Bryant Stats, Video, Bio, Profile**
   www.nba.com/playerfile/kobe_bryant/
   $p_3 = \texttt{www.nba.com/playerfile/*/}$
4. Kobe Bryant Biography
   www.biography.com/people/kobe-bryant-10683945
   $p_4 = \texttt{www.biography.com/people/*}$
5. **(Clicked) Kobe Bryant — Los Angeles Lakers**
   sports.yahoo.com/nba/players/3118
   $p_5 = \texttt{sports.yahoo.com/nba/players/*}$

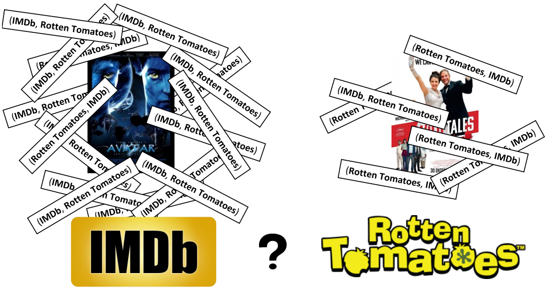
Then we have: $(p_3, p_1), (p_3, p_4), (p_5, p_1), (p_5, p_4)$

More Details about Preference Extraction

- Consider only the SAT-Clicks
    - satisfied clicks with $\geq$ 30 seconds dwell time [Wang *et al.*, KDD'09]
- Focus on web pages of **the first page** of search results

Drawback

- Biased by **popular queries** or **popular entities**



Preference Aggregation

- **aggregate preferences** by entities to avoid the popularity bias
- estimate the **probability** of observing the preference of two patterns
  - (p_1, p_2, w) denotes $P(\text{the pattern } p_1 \text{ is more relevant than } p_2) = w$.

(IMDb, Rotten Tomatoes) x 8000 → **(IMDb, Rotten Tomatoes, 0.8)**

(Rotten Tomatoes, IMDb) x 2000 → **(Rotten Tomatoes, IMDb, 0.2)**

(IMDb, Rotten Tomatoes) x 30 → **(IMDb, Rotten Tomatoes, 0.3)**

(Rotten Tomatoes, IMDb) x 70 → **(Rotten Tomatoes, IMDb, 0.7)**

Finally, we have many weighted pairwise preferences for each type.

Relevance Optimization for $P(p \mid t)$

- A pairwise preference can be treated as a partial constraint.
- Assume $P(p \mid t)$ can be represented by a logistic function:

$$P\left(p_i \mid t\right) = \frac{1}{1 + \exp\left(-\theta_{i,t}\right)}$$

  - $\theta_{i,t}$ is a parameter for p_i and the type t.

Relevance Optimization for $P(p \mid t)$ (Cont'd)

- Adopt the RankNet cost function to optimize pairwise preference

$$\sum_{e \in E(t)} \sum_{(p_1, p_2, w) \in R_e} w \cdot \log\left(1 + \exp\left(P\left(p_2 \mid t\right) - P\left(p_1 \mid t\right)\right)\right)$$

  - R_e is the list of weighted preferences for the entity e.
  - $P\left(p_i \mid t\right)$ is the relevance using current parameters $\theta_{i,t}$.
- The gradient descent method is applied for optimization.

Type Distribution $P(t \mid q)$

- Estimate how much users treat q is a query of the type t.
- Adopt click-through data and Bayesian m-estimate smoothing

$$P(t \mid q) = \frac{\sum_{p \in S} P(t \mid p) \cdot Click(p, q) + m \cdot P(t)}{m + \sum_{t \in T(q)} \sum_{p \in S} P(t \mid p) \cdot Click(p, q)}$$

  - S is the set containing all patterns.
  - $P(t)$ can be computed by normalizing the number of entities.

Type Distribution $P(t \mid q)$ (Cont'd)

- $P(t \mid p)$ can be calculated with the Bayes' theorem as follows

$$P(t \mid p) = \frac{P(p \mid t) \cdot P(t)}{P(p)}$$

  - $P(p \mid t)$ $P(p \mid t)$ is pattern-type relevance.
  - $P(p)$ can be computed by normalizing clicks in logs.

Parameter Optimization

- For multiple parameters, we would like to learn β parameters.
- Use the RankNet cost function for optimization

$$\sum_{q \in Q} \sum_{(u_1, u_2) \in R(q)} \log \left(1 + \exp \left(P \left(u_2 \mid q, i_{u_2}\right) - P \left(u_1 \mid q, i_{u_1}\right)\right)\right),$$

  - $R(q)$ is a set of preferences for URLs from q's click-though data.
  - i_u is the ranked position of u in the original search result.

Future Work

- Supervised Approach
  - Directly apply the ranking consistency into learning to rank
  - Optimize the ranking consistency and cost function at the same time
- Unsupervised Approach
  - Consider the ranking consistency while calculating retrieval functions
  - Not only compute simple measures, but also leverage other queries

Evaluation of Pattern-Type Relevance

- To estimate pattern-type relevance $P(p \mid t)$ is the key of the model.
- Totally 107,531 URL patterns are extracted.
- Collect the top five patterns for each type for evaluation
- Baseline is a frequency-based model by clicked counts.
- Use NDCG@k as the evaluation measure
- Hire two assessors to manually judge collected URL patterns
- Three kinds of relevance scores
  - Relevant and important (Score 5) - `ESPN.com` to athletes
  - Generally relevant (Score 1) - `Biography.com` to athletes
  - Irrelevant (Score 0) - `IMDb` to athletes

Evaluation of Pattern-Type Relevance (Cont'd)

- All collected patterns are annotated by two assessors.
  - 80.76% agreement with 0.65 unweighted kappa coefficient

Measure	Frequency	Our Approach
NDCG@1	0.9607	0.9821 (+2.23%)
NDCG@2	0.7655	0.8145 (+5.87%)
NDCG@3	0.6748	0.7363 (+8.61%)
NDCG@4	0.6267	0.6800 (+8.07%)
NDCG@5	0.5857	0.6450 (+9.69%)

- Our approach significantly outperforms the baseline.
- The baseline is biased by popular entities.