

RANKING USING MULTIPLE DOCUMENT TYPES IN DESKTOP SEARCH

Jinyoung Kim and W. Bruce Croft

CIIR, UMass Amherst

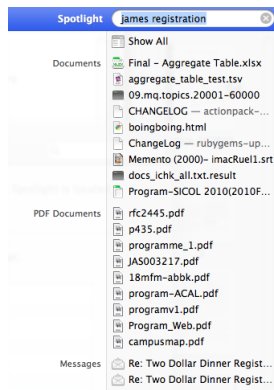
Yahoo! Research Barcelona

Outline

- 1 Introduction
- 2 Retrieval Model
 - Type-specific Retrieval Models
 - Type-prediction Methods
 - Summary
- 3 Test Collection Generation Methods
- 4 Experiments
- 5 Conclusions
- 6 Appendix

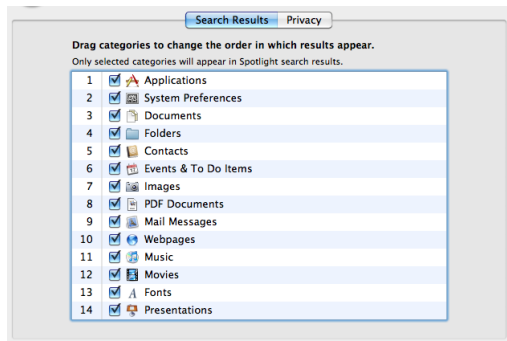
Motivating Example

Query : James Registration



Motivating Example

Can't we do better than this?



Desktop Search

Significance

- Most common search system for personal information

Characteristics

- People mostly do 're-finding' [Elaweil07]
 - Known-item search
- Many document types
 - Meta-search [Thomas09]
- Unique metadata for each type
 - Semi-structured document retrieval (TREC Email Track)

This characterization holds true in the age of cloud computing!

Past Works on Desktop Search

Focuses

- User interface issues [Dumais03,06]
- Desktop-specific features [Solus06] [Cohen08]

Limitations

- Each based on different user study
- None of them performed a comparative evaluation

Pseudo-desktop Method [Kim09]

- TREC-style evaluation for desktop search

Trading external validity for experimental control & repeatability

Contributions

Document Type Prediction Method

- Show the impact of type prediction method to the final ranking
- New type prediction method that exploits document metadata
- Combination of evidence improves the performance further

Evaluation

- Game interface to collect a large amount of human queries

Outline

- 1 Introduction
- 2 Retrieval Model
 - Type-specific Retrieval Models
 - Type-prediction Methods
 - Summary
- 3 Test Collection Generation Methods
- 4 Experiments
- 5 Conclusions
- 6 Appendix

Retrieval Model for Desktop Search

Type-specific Ranking

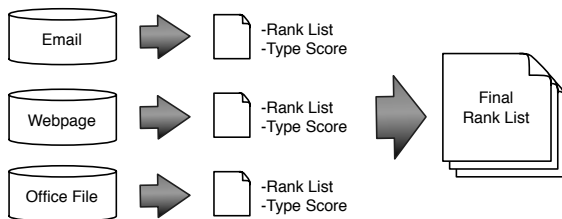
- Use the most suitable ranking algorithm

Type Prediction

- Predict which document type the user is looking for

Merge into Final Result

- Rank list merging



Type-specific Retrieval Models

Retrieval Models:

- DQL : Document Query Likelihood
- PRM-S : Probabilistic Retrieval Model for Semistructured Data [Kim09]
- PRM-D : Interpolation of DQL and PRM-S
 - More stable performance than PRM-S

Notations:

- Query $Q = (q_1, \dots, q_m)$
- Collection C with fields (F_1, \dots, F_n)
- Each document d with fields (f_1, \dots, f_n)

Type-specific Retrieval Models

Retrieval Models:

- DQL : Document Query Likelihood
- PRM-S : Probabilistic Retrieval Model for Semistructured Data [Kim09]
- PRM-D : Interpolation of DQL and PRM-S
 - More stable performance than PRM-S

Notations:

- Query $Q = (q_1, \dots, q_m)$
- Collection C with fields (F_1, \dots, F_n)
- Each document d with fields (f_1, \dots, f_n)

Probabilistic Retrieval Model for Semistructured Data [Kim09]

Basic Idea:

- Estimate the implicit mapping of each query word to document fields
 - e.g. james (\rightarrow *sender*) registration (\rightarrow *title*) 2010 (\rightarrow *date*)
- Combine field-level evidences based on mapping probability $P_M(F_j|q_i)$

$$P(Q|d) = \prod_{i=1}^m \sum_{j=1}^n P_M(F_j|q_i) P_{QL}(q_i|f_j) \quad (1)$$

- Compare with mixture model using fixed field weights w_j

$$P(Q|d) = \prod_{i=1}^m \sum_{j=1}^n w_j P_{QL}(q_i|f_j) \quad (2)$$

Performance for TREC email collection: [Kim09]

Collection	DQL	MFLM	BM25F	PRM-S	PRM-D
TREC	0.538	0.559	0.594	0.617	0.624

Probabilistic Retrieval Model for Semistructured Data [Kim09]

Basic Idea:

- Estimate the implicit mapping of each query word to document fields
 - e.g. james (\rightarrow *sender*) registration (\rightarrow *title*) 2010 (\rightarrow *date*)
- Combine field-level evidences based on mapping probability $P_M(F_j|q_i)$

$$P(Q|d) = \prod_{i=1}^m \sum_{j=1}^n P_M(F_j|q_i) P_{QL}(q_i|f_j) \quad (1)$$

- Compare with mixture model using fixed field weights w_j

$$P(Q|d) = \prod_{i=1}^m \sum_{j=1}^n w_j P_{QL}(q_i|f_j) \quad (2)$$

Performance for TREC email collection: [Kim09]

Collection	DQL	MFLM	BM25F	PRM-S	PRM-D
TREC	0.538	0.559	0.594	0.617	0.624

Probabilistic Retrieval Model for Semistructured Data [Kim09]

Basic Idea:

- Estimate the implicit mapping of each query word to document fields
 - e.g. james (\rightarrow *sender*) registration (\rightarrow *title*) 2010 (\rightarrow *date*)
- Combine field-level evidences based on mapping probability $P_M(F_j|q_i)$

$$P(Q|d) = \prod_{i=1}^m \sum_{j=1}^n P_M(F_j|q_i) P_{QL}(q_i|f_j) \quad (1)$$

- Compare with mixture model using fixed field weights w_j

$$P(Q|d) = \prod_{i=1}^m \sum_{j=1}^n w_j P_{QL}(q_i|f_j) \quad (2)$$

Performance for TREC email collection: [Kim09]

Collection	DQL	MFLM	BM25F	PRM-S	PRM-D
TREC	0.538	0.559	0.594	0.617	0.624

Improving Type Prediction Method

Query-likelihood of Collection [Si02]

- Match each query-term to the collection LM
- Best performance in recent evaluation [Thomas09]

$$CQL(Q, C) = \prod_{q \in Q} P(q|C) \quad (3)$$

Field-based collection query likelihood (FQL)

- Match each query-term to field-level collection LM
 - e.g. james (\rightarrow *sender*) registration (\rightarrow *title*) 2010 (\rightarrow *date*)
- Combine field-level scores into a collection score

$$FQL(Q, C) = \prod_{q \in Q} \text{avg}_{F_i \in C} (P(q|F_i)) \quad (4)$$

Improving Type Prediction Method

Query-likelihood of Collection [Si02]

- Match each query-term to the collection LM
- Best performance in recent evaluation [Thomas09]

$$CQL(Q, C) = \prod_{q \in Q} P(q|C) \quad (3)$$

Field-based collection query likelihood (FQL)

- Match each query-term to field-level collection LM
 - e.g. james (\rightarrow *sender*) registration (\rightarrow *title*) 2010 (\rightarrow *date*)
- Combine field-level scores into a collection score

$$FQL(Q, C) = \prod_{q \in Q} \text{avg}_{F_i \in C} (P(q|F_i)) \quad (4)$$

More Type-prediction Methods

Name	Source	Remark
QL of Collection (CQL)	LM of collection	Terms similar to each collection will be used in a query
QL of Field (FQL)	LM of collection fields	Similar to CQL, yet use field-level evidences
Dictionary-matching QL of Query Log (QQL)	List of matching words LM of query logs	e.g. meeting → email Terms similar to previous queries will be used
Geometric Average	TopK Document Scores	Top documents can represent the collection
ReDDE	TopK Document Scores	Model the expected count of the relevant documents
Query Clarity	LM of collection	Model the expected performance of query for each collection

Combining Type Prediction Methods

Grid-search of Parameter Values (Grid)

- Iteratively find the best-performing parameter setting

Multi-class Classifier (MultiSVM)

- Cast as an one-versus-rest classification

Rank-learning Method (RankSVM)

- Cast as a collection ranking problem
- The collection containing relevant documents is preferred to others

Retrieval Model for Desktop Search (summary)

Type-specific Ranking

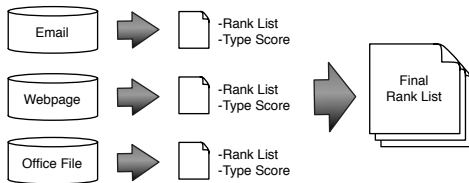
- DQL / PRM-S / PRM-D
- **Best** : use the best-performing method for each type

Type Prediction

- Features : CQL / FQL / Dict / QQL / Clarity / GAVG / ReDDE
- Combination : Grid / MultiSVM / RankSVM
- **Oracle** : always knows correct type

Merge into Final Result [Callan95]

- CORI algorithm



Outline

- 1 Introduction
- 2 Retrieval Model
 - Type-specific Retrieval Models
 - Type-prediction Methods
 - Summary
- 3 Test Collection Generation Methods
- 4 Experiments
- 5 Conclusions
- 6 Appendix

Pseudo-desktop Method [Kim09]

Procedure

- Collect documents of reasonable size and variety
 - Emails from the mailing list
 - Office documents by web search API
- Generate queries by taking terms from the target document
(adapted from [Azzopardi07])

Query	Target Collection
jose 03 kahan	email
presentation 19	ppt
org address	html

- Manual queries were collected and used to validate generated queries
 - By showing documents and asking for hypothetical queries

DocTrack – Using HCG to collect known-item queries

Basic Idea

- 1 The user is shown a target document for 15 seconds
- 2 The user is asked to find the document
- 3 Score is given using the position of target document found

DocTrack Game

Home Start Scoreboard Logout (logged in as babo)

Here's the 2nd document ([skip to search](#))

Type: file

Title: cs/www.cs.umass.edu/~yungchih/publication06_CHANTS_HEC.pdf (query)

URL: http://lifedee.cs.umass.edu/craw/ecs/~yungchih/publication06_CHANTS_HEC.pdf

Metadata:

filename:cs/www.cs.umass.edu/~yungchih/publication06_CHANTS_HEC.pdf

Content:

(click [here](#) to see the content if it's invisible)

1 / 8

A Hybrid Routing Approach for Opportunistic Networks

Ling-Jyh Chen Chen-Hung Yu Tony Sun Yung-Chih Chen Hao-hua Chu
 Academia Sinica National Taiwan University UCLA Academia Sinica National Taiwan University

cs@lifedee.cs.umass.edu y99523@statu.edu.tw tong@cs.ucla.edu ych@lifedee.cs.umass.edu hcc@lifedee.cs.umass.edu

ABSTRACT

With wireless networking technologies embedding into the fabric of our working and operating environments, proper handling of intermittent wireless connectivity and network disruption is of significant concern. As the above number of pub-

1. INTRODUCTION

With prevalent adoption and usage of wireless communication technologies, networking challenges evolve when continuous network connectivity cannot be guaranteed. For these challenging networking environments such as found in

Status

Pages Found : 8/10
 Queries Issued : 0/5
 Current Score : 0.0

00:00

Top Scorers

uyssal : 79 (2010-01-07)
 uysal : 67 (2010-01-05)
 uysal : 53 (2010-01-08)
 jangwon : 52 (2010-01-07)
 uysal : 45 (2010-01-07)
 yungah : 39 (2010-01-06)
 uysal : 36 (2010-01-08)
 jangwon : 36 (2010-01-07)
 sjh : 35 (2010-01-05)
 uysal : 33 (2010-01-07)
 yungah : 32 (2010-01-06)

Frequent Players

uyssal : 6 times (avg: 52)
 yungah : 3 times (avg: 34)
 sjh : 2 times (avg: 30)
 jangwon : 2 times (avg: 44)

DocTrack Game (cont.)

Modifying of PageHunt game for Desktop Search [Ma09]

- Collect documents participants are familiar with
 - CS Collection : department emails, calendar items and documents
- Users are shown multiple target documents
 - Simulate the vague memory of known-item searcher

Query	Target Collection
reminder jeffrey johns	email
2010 candidate weekend	calendar item
yanlei xml dissemination	office document
cs646 homework html	html

Outline

- 1 Introduction
- 2 Retrieval Model
 - Type-specific Retrieval Models
 - Type-prediction Methods
 - Summary
- 3 Test Collection Generation Methods
- 4 Experiments**
- 5 Conclusions
- 6 Appendix

Experimental Setting

Collections

- Three Pseudo-desktop Collections
 - Automatically generated queries
 - 100 queries / average length 2 (sd : 1)
- CS Collection
 - Human-formulated queries from Doctrack game (984)
 - 984 queries / average length 3.97 (sd : 1.85)

Other details

- Reciprocal Rank was used to evaluate retrieval results

Collection Statistics

Pseudo-desktop Collections

Type	Jack		Tom		Kate	
email	6067	(555)	6930	(558)	1669	(935)
html	953	(3554)	950	(3098)	957	(3995)
pdf	1025	(8024)	1008	(8699)	1004	(10278)
doc	938	(6394)	984	(7374)	940	(7828)
ppt	905	(1808)	911	(1801)	729	(1859)

CS Collection

Type	#Docs	Length
email	851	(731)
news article	170	(352)
calendar item	354	(306)
webpage	4727	(539)
office document	1887	(357)

Result - Type Prediction Accuracy

Pseudo-desktop Collections

	Jack	Tom	Kate
CQL	0.606	0.637	0.38
FQL	0.773	0.807	0.64

CS Collection

Method	CQL	FQL	Grid	RankSVM	MultiSVM
Accuracy	0.708	0.743	0.747	0.758	0.808

- FQL improves performance over CQL
- Combination methods improve the performance further

Result - Type Prediction Accuracy

Method	CQL	FQL	Dict	QQL	Clarity	GAVG	ReDDE
Single-feature Accuracy	0.708	0.743	0.201	0.579	0.240	0.255	0.207
Leave-one-out Accuracy	-0.6%	-1.7%	-0.6%	-3.1%	-0.6%	-0.0%	-0.0%

- FQL has the best performance among features
- QQL has the most impact when left out
- Features based on document scores were not effective in general

Result - Retrieval Performance

Pseudo-desktop Collections

	CQL	Jack FQL	Oracle
DQL	0.159	0.27	0.331
PRM-S	0.212	0.326	0.403
PRM-D	0.219	0.335	0.403
Best	0.225	0.336	0.414

CS Collection

	CQL	FQL	Grid	RankSVM	MultiSVM	Oracle
DQL	0.507	0.53	0.552	0.563	0.556	0.674
PRM-S	0.501	0.518	0.518	0.551	0.547	0.674
PRM-D	0.518	0.536	0.536	0.567	0.564	0.694
Best	0.548	0.564	0.590	0.596	0.594	0.72

Result - Retrieval Performance

Pseudo-desktop Collections

	CQL	Jack FQL	Oracle
DQL	0.159	0.27	0.331
PRM-S	0.212	0.326	0.403
PRM-D	0.219	0.335	0.403
Best	0.225	0.336	0.414

CS Collection

	CQL	FQL	Grid	RankSVM	MultiSVM	Oracle
DQL	0.507	0.53	0.552	0.563	0.556	0.674
PRM-S	0.501	0.518	0.518	0.551	0.547	0.674
PRM-D	0.518	0.536	0.536	0.567	0.564	0.694
Best	0.548	0.564	0.590	0.596	0.594	0.72

Outline

- 1 Introduction
- 2 Retrieval Model
 - Type-specific Retrieval Models
 - Type-prediction Methods
 - Summary
- 3 Test Collection Generation Methods
- 4 Experiments
- 5 Conclusions**
- 6 Appendix

Conclusions & Future Works

Conclusions

- Field-based collection query likelihood (FQL) shows superior accuracy
- Combination can improve type prediction performance further
- Game-based method can be used to evaluate retrieval models

Future Works

- Type-specific Ranking
 - Exploit unique features for each collection
- Type Prediction Model
 - More features
- DocTrack game
 - Study session-level data

Conclusions & Future Works

Conclusions

- Field-based collection query likelihood (FQL) shows superior accuracy
- Combination can improve type prediction performance further
- Game-based method can be used to evaluate retrieval models

Future Works

- Type-specific Ranking
 - Exploit unique features for each collection
- Type Prediction Model
 - More features
- DocTrack game
 - Study session-level data

Thank you for your attention!

Any questions?

Outline

- 1 Introduction
- 2 Retrieval Model
 - Type-specific Retrieval Models
 - Type-prediction Methods
 - Summary
- 3 Test Collection Generation Methods
- 4 Experiments
- 5 Conclusions
- 6 Appendix**

Other Type-prediction Methods

Query-likelihood of Query Log

$$QQL(Q, C) = \prod_{q \in Q} P(q|L_C) \quad (5)$$

Geometric Mean of Top Document Scores [Seo08]

$$GAVG(Q, C) = \left(\prod_{d \in D_{top}} P(Q|d) \right)^{\frac{1}{m}} \quad (6)$$

ReDDE [Si03]

$$ReDDE(Q, C) = \sum_{d \in D_{top}} P(Q|d) \quad (7)$$

Query Clarity [Cronen-Townsend02]

$$Clarity(Q, C) = \sum_{w \in V} P(w|L_Q) \log_2 \frac{P(w|L_Q)}{P(w|C)} \quad (8)$$

CORI Algorithm for Rank-list Merging

$$C'_i = (C_i - C_{min}) / (C_{max} - C_{min}) \quad (9)$$

$$D' = (D - D_{min}) / (D_{max} - D_{min}) \quad (10)$$

$$D'' = \frac{D' + 0.4 \cdot D' \cdot C'_i}{1.4} \quad (11)$$