

# RETRIEVAL EXPERIMENTS USING PSEUDO-DESKTOP COLLECTIONS

Jinyoung Kim and W. Bruce Croft

CIIR, UMass Amherst

CIKM'09

# Outline

- 1 Introduction
- 2 Building Pseudo-desktop Collections
- 3 Retrieval Models for Desktop Search
- 4 Retrieval Experiments
- 5 Conclusions

# Desktop Search

## Significance

- Most common search system for personal information

## Characteristics & Relevant Problems

- People mostly do 're-finding'
  - Known-item search
- Many document types
  - Meta-search
- Unique metadata for each type
  - Semi-structured document retrieval

This definition holds true even for 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 comparative evaluation

**Table:** Statistics of desktop collections from previous research

Previous Work	#Desktops	#Files	Query Length	Document Types
Dumais et al.	225	36182	1.6	e-mails: 80%
Chernov et al.	14	3433	1.7	e-mails : 82.7%
Cohen et al.	19	N/A	N/A	documents: 41.2%

Key missing piece here is a reusable collection!

# Contributions

## Pseudo-desktop Collections

- Suggested a new query generation method with higher validity
- Suggested a new way to evaluate the validity

## Retrieval Experiments

- Compared semi-structured document retrieval methods
- Improved and analyzed the performance of PRM-S [Kim09]

# Outline

- 1 Introduction
- 2 Building Pseudo-desktop Collections**
- 3 Retrieval Models for Desktop Search
- 4 Retrieval Experiments
- 5 Conclusions

# Collecting Documents

## Criteria

- Reasonable size and variety
- Availability of metadata
- No privacy concern

## Procedure

- Filter public email collection by person name
- Add office/web documents by web search for each person's profile

**Table:** Statistics of three 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)

# Generating Queries

## Assumption

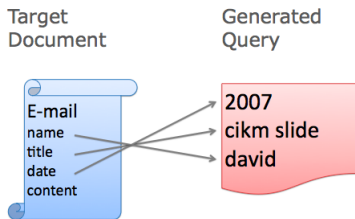
- Users take query terms from the target document [Azzopardi07]

## Procedure

- Choose a target document
- Choose the **extent** of term selection
- Choose **term** from the extent based on some distribution

## Parameters

- Choice of **extent** : Document vs. Field
- Choice of **term** : Uniform / TF / IDF / TF\*IDF



# Validating Generating Queries

## By Comparison with Manual Queries

- Compare Query-terms (Predictive Validity)
  - Generation probability  $P_{term}(Q)$  of the manual query  $Q$

$$P_{term}(Q) = \prod_{q_i \in Q} P_{term}(q_i) \quad (1)$$

- Compare the Distribution of Retrieval Scores (Replicative Validity)
  - Two-sided Kolmogorov-Smirnov test [Azzopardi07]

## Evaluation Settings:

- W3C mailing list (200k docs)
  - Manual : 150 TREC known-item queries
  - Generated : 1000 queries for each method

# Validating Generating Queries

## By Comparison with Manual Queries

- Compare Query-terms (Predictive Validity)
  - Generation probability  $P_{term}(Q)$  of the manual query  $Q$

$$P_{term}(Q) = \prod_{q_i \in Q} P_{term}(q_i) \quad (1)$$

- Compare the Distribution of Retrieval Scores (Replicative Validity)
  - Two-sided Kolmogorov-Smirnov test [Azzopardi07]

## Evaluation Settings:

- W3C mailing list (200k docs)
  - Manual : 150 TREC known-item queries
  - Generated : 1000 queries for each method

# Evaluation of Predictive Validity

Table: Sum of generation probabilities for different generation methods

Extent : Document		Extent : Field	
$P_{term}$	$\sum \log P_{term}(Q)$	$P_{term}$	$\sum \log P_{term}(Q)$
Uniform	-26.457	Uniform	<b>-23.460</b>
TF	-22.782	TF	<b>-22.394</b>
IDF	-21.876	IDF	<b>-17.990</b>
TF*IDF	-18.269	TF*IDF	<b>-17.180</b>

- Field-based extent selection results in higher validity

# Evaluation of Replicative Validity

Table: p-values of KS-test against the score distribution of manual queries

Extent	$P_{term}$	DLM	PRM-S	PRM-D	Average
Document	Uniform	0.003	0.000	0.041	0.015
	TF	0.090	0.000	0.005	0.032
	IDF	0.000	0.023	0.000	0.008
	TF*IDF	0.000	<b>0.160</b>	0.000	0.053
Average		0.023	0.046	0.012	0.027
Field	Uniform	<b>0.085</b>	<b>0.323</b>	<b>0.276</b>	<b>0.228</b>
	TF	<b>0.105</b>	<b>0.667</b>	<b>0.570</b>	<b>0.447</b>
	IDF	<b>0.068</b>	0.013	0.008	0.030
	TF*IDF	<b>0.284</b>	0.021	0.022	0.109
Average		0.136	0.256	0.219	0.204

- Field-based extent selection results in higher validity

# Outline

- 1 Introduction
- 2 Building Pseudo-desktop Collections
- 3 Retrieval Models for Desktop Search**
- 4 Retrieval Experiments
- 5 Conclusions

# Retrieval Models

## Notations:

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

## Retrieval Models:

- Document Query Likelihood (DLM)
- Mixture of Field Language Models (MFLM) [Ogilvie03]

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

- BM25F [Robertson04]

$$weight(q_i, d) = \sum_{f_j \in d} \frac{w_j \times tf(q_i, f_j)}{(1 - b_j) + b_j \times \frac{|f_j|}{|F_j|}} \quad (3)$$

# Retrieval Models

Notations:

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

Retrieval Models:

- Document Query Likelihood (DLM)
- Mixture of Field Language Models (MFLM) [Ogilvie03]

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

- BM25F [Robertson04]

$$weight(q_i, d) = \sum_{f_j \in d} \frac{w_j \times tf(q_i, f_j)}{(1 - b_j) + b_j \times \frac{|f_j|}{|F_j|}} \quad (3)$$

## Retrieval Models (cont.)

PRM-S (Probabilistic Retrieval Model for Semistructured Data [Kim09])

- Probabilistically map each query word with document field
  - e.g. Meg (cast:0.8) Ryan (cast:0.7) Romance (genre:0.6)
- Combine field LMs based on mapping probability

$$P(Q|d) = \prod_{i=1}^m \sum_{j=1}^n P_M(F_j|q_i) P_{QL}(q_i|f_j) \quad (4)$$

Improving Field-level Score Estimation of PRM-S

- Mixture of DLM and PRM-S (PRM-D)
- 2-Stage Dirichlet Smoothing (PRM-S2) [Zhao08]

2-stage smoothing provides better control at the cost of more parameters

## Retrieval Models (cont.)

PRM-S (Probabilistic Retrieval Model for Semistructured Data [Kim09])

- Probabilistically map each query word with document field
  - e.g. Meg (cast:0.8) Ryan (cast:0.7) Romance (genre:0.6)
- Combine field LMs based on mapping probability

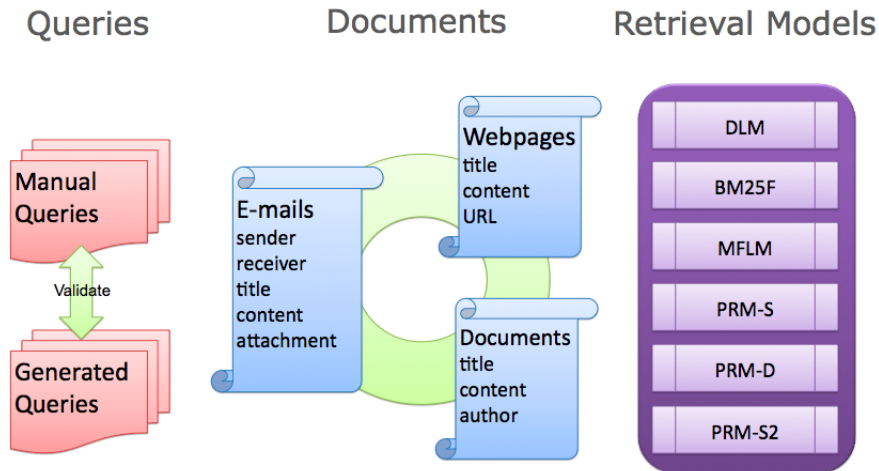
$$P(Q|d) = \prod_{i=1}^m \sum_{j=1}^n P_M(F_j|q_i) P_{QL}(q_i|f_j) \quad (4)$$

Improving Field-level Score Estimation of PRM-S

- Mixture of DLM and PRM-S (PRM-D)
- 2-Stage Dirichlet Smoothing (PRM-S2) [Zhao08]

2-stage smoothing provides better control at the cost of more parameters

# Big Picture



# Outline

- 1 Introduction
- 2 Building Pseudo-desktop Collections
- 3 Retrieval Models for Desktop Search
- 4 Retrieval Experiments**
- 5 Conclusions

# Experimental Setting

## Collections

- W3C mailing list
  - TREC queries (25 train / 125 test)
- Pseudo-desktop
  - Manual queries (50)
    - Collected by showing people documents and asking queries
  - Generated queries (100)
    - Field-based method, with average length 2

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)

# Experimental Setting (cont.)

## Query Examples

TREC Manual Queries
Preliminary Report from the WSDL Attributes Task Force XML DSig 99 MobiQuitous 2004 latest deadlines
Pseudo-desktop Manual Queries
Martyn Jan OCLC proof checking syntax for RDF
Pseudo-desktop Generated Queries
jose 03 kahan other ua amaya 48

## Other details

- All parameters tuned using TREC training queries
- Reciprocal Rank was used as the evaluation measure

## Result - Manual Queries

**Table:** Retrieval performance for TREC email collection

Collection	DLM	MFLM	BM25F	PRM-S	PRM-D	PRM-S2
TREC	0.538	0.559	0.594	0.617	0.619	<b>0.630</b>

- $PRM-S < PRM-D < PRM-S2$
- PRM-S2 outperforms the best TREC submission (0.621)

**Table:** Retrieval performance for pseudo-desktop email collections

Collection	DLM	MFLM	BM25F	PRM-S	PRM-D	PRM-S2
PD-Jack	0.378	0.235	0.229	0.334	<b>0.389</b>	0.356
PD-Tom	0.403	0.312	0.311	0.422	<b>0.457</b>	0.438
PD-Kate	<b>0.482</b>	0.307	0.401	0.413	0.463	0.455

- $PRM-S < PRM-S2 < PRM-D$

## Result - Generated Queries in Pseudo-desktop

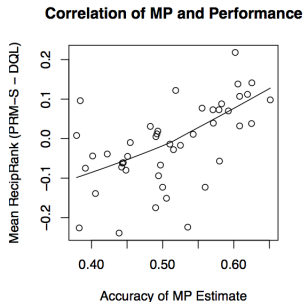
Type	User	DLM	PRM-S	PRM-D
email	Jack	0.213	0.285	0.275
	Tom	0.197	0.244	0.243
	Kate	0.329	0.438	0.413
Average		0.246	<b>0.323</b>	0.310
html	Jack	0.418	0.428	0.440
	Tom	0.428	0.381	0.409
	Kate	0.422	0.377	0.410
Average		<b>0.423</b>	0.395	0.419
pdf	Jack	0.417	0.378	0.410
	Tom	0.411	0.359	0.392
	Kate	0.408	0.420	0.453
Average		0.412	0.385	<b>0.419</b>

### Result

- DLM is better for html, while PRM-S is better for email
- PRM-D shows consistently high performance

# Analysis : What Makes PRM-S Better?

The accuracy of mapping probability



The field from which query terms are chosen

Type	Field	DLM	PRM-S	PRM-D
email	title	0.251	<b>0.389</b>	0.342
	content	<b>0.339</b>	0.267	0.327
html	title	0.461	0.533	<b>0.547</b>
	content	<b>0.514</b>	0.278	0.339

# Outline

- 1 Introduction
- 2 Building Pseudo-desktop Collections
- 3 Retrieval Models for Desktop Search
- 4 Retrieval Experiments
- 5 Conclusions

# Conclusions & Future Works

## Conclusions

- Field-based query generation is more valid
  - Insight into querying behavior of known-item search
- PRM-S is effective and PRM-D / PRM-S2 improves it further
  - Different collection requires different retrieval model
- Generated queries are useful to analyze the performance
  - Enables completely controlled experiments

## Future Works

- Pseudo-desktop
  - More realistic query generation and evaluation method
  - Further validation by user study
- Retrieval Model
  - Merging the results from each sub-collecton

# Conclusions & Future Works

## Conclusions

- Field-based query generation is more valid
  - Insight into querying behavior of known-item search
- PRM-S is effective and PRM-D / PRM-S2 improves it further
  - Different collection requires different retrieval model
- Generated queries are useful to analyze the performance
  - Enables completely controlled experiments

## Future Works

- Pseudo-desktop
  - More realistic query generation and evaluation method
  - Further validation by user study
- Retrieval Model
  - Merging the results from each sub-collecton