Learning User Reformulation Behavior for Query Auto-Completion

Jyun-Yu Jiang, Yen-Yu Ke, Pao-Yu Chien and Pu-Jen Cheng

Department of Compute Science and Information Engineering

National Taiwan University

July 9, 2014 (SIGIR)

Query Auto-Completion (QAC)

- A common feature in modern search engines
  - Help users formulate queries while typing in the search boxes
- Given a user-typed prefix, N ranked completions are shown

Why Query Auto Completion?

- Typing queries costs too much
  - Users can save their keystrokes
- Further benefits
  - Spelling errors, query expansion, speed, ...

The goal of QAC

Rank the user's intended query in a high position
with as few keystrokes as possible

Context-Aware Approaches

- Context captures user's search intents.
  - submitted queries
  - click-through information

$$\underbrace{q_1 \rightarrow q_2 \rightarrow \cdots \rightarrow \cdot q_{T-1}}_{context} \rightarrow q_T$$

- Previous work statistically models query dependencies and similarity.

Query Session

- query dependencies [He2009]
- query similarity [Bar-Yossef2011]
- personal history [Shokouhi2013]

Click-through Data

- relevant queries [Mei2009]
- query clusters [Liao2011]
- click behavior [Ozertem2012]

However, a user may have some behavior in the context.

Context-Aware Approaches

- Context captures user's search intents.
  - submitted queries
  - click-through information

$$\underbrace{q_1 \to q_2 \to \cdots \to \cdot q_{T-1}}_{context} \to q_T$$

- Previous work statistically models query dependencies and similarity.

Query Session

- query dependencies [He2009]
- query similarity [Bar-Yossef2011]
- personal history [Shokouhi2013]

Click-through Data

- relevant queries [Mei2009]
- query clusters [Liao2011]
- click behavior [Ozertem2012]

However, a user may have some behavior in the context.

Example Completions

$$\underbrace{\textit{“stomach sounds”} \rightarrow \textit{“irritable bowel syndrome”}}_{\textit{Context}} \rightarrow \underbrace{\textit{“cramps stomach”}}_{\textit{Intended Query}}$$

Completions from Conventional Approaches

- *“colon cancer symptoms”* (query similarity/dependencies)
  - from a context-aware QAC approach [Bar-Yossef *et al.*, 2011]
- *“celiac disease”* (query dependencies)
  - from a context-aware QS approach [He *et al.*, 2009]
- *“colon cancer”* (query clusters)
  - from a cluster-based context-aware QS approach [Liao *et al.*, 2011]

How users reformulate their queries in search sessions?

Example Completions

$$\underbrace{\text{“\textit{stomach sounds}”} \rightarrow \text{“\textit{irritable bowel syndrome}”}}_{\textit{Context}} \rightarrow \underbrace{\text{“\textit{cramps stomach}”}}_{\textit{Intended Query}}$$

Completions from Conventional Approaches

- *“colon cancer symptoms”* (query similarity/dependencies)
  - from a context-aware QAC approach [Bar-Yossef *et al.*, 2011]
- *“celiac disease”* (query dependencies)
  - from a context-aware QS approach [He *et al.*, 2009]
- *“colon cancer”* (query clusters)
  - from a cluster-based context-aware QS approach [Liao *et al.*, 2011]

How users reformulate their queries in search sessions?

User Reformulation Behavior

- Studied as query reformulation strategies [Huang *et al.*, 2009].

Semantic Relations [Akahani *et al.*, 2002] – Difficult to Analyze

- *specialization*: narrow the search constraints, e.g., *computer* $\rightarrow$ *mac*
- *generalization*: relax the search constraints, e.g., *lion* $\rightarrow$ *animal*

Syntactic Relations [Rieh *et al.*, 2006] – Simple to Analyze

- Syntactic and explicit changes between queries
  - Such as adding terms, removing terms, acronym expansion.
- Clear definitions of reformulation types [Jansen *et al.*, 2009]
- Personalization [Jiang *et al.*, 2011]

Can we exploit user reformulation behavior to QAC?

User Reformulation Behavior

- Studied as query reformulation strategies [Huang *et al.*, 2009].

Semantic Relations [Akahani *et al.*, 2002] – Difficult to Analyze

- *specialization*: narrow the search constraints, e.g., *computer* → *mac*
- *generalization*: relax the search constraints, e.g., *lion* → *animal*

Syntactic Relations [Rieh *et al.*, 2006] – Simple to Analyze

- Syntactic and explicit changes between queries
  - Such as adding terms, removing terms, acronym expansion.
- Clear definitions of reformulation types [Jansen *et al.*, 2009]
- Personalization [Jiang *et al.*, 2011]

Can we exploit user reformulation behavior to QAC?

User Reformulation Behavior

- Studied as query reformulation strategies [Huang *et al.*, 2009].

Semantic Relations [Akahani *et al.*, 2002] – Difficult to Analyze

- *specialization*: narrow the search constraints, e.g., *computer* → *mac*
- *generalization*: relax the search constraints, e.g., *lion* → *animal*

Syntactic Relations [Rieh *et al.*, 2006] – Simple to Analyze

- Syntactic and explicit changes between queries
  - Such as adding terms, removing terms, acronym expansion.
- Clear definitions of reformulation types [Jansen *et al.*, 2009]
- Personalization [Jiang *et al.*, 2011]

Can we exploit user reformulation behavior to QAC?

User Reformulation Behavior

- Studied as query reformulation strategies [Huang *et al.*, 2009].

Semantic Relations [Akahani *et al.*, 2002] – Difficult to Analyze

- *specialization*: narrow the search constraints, e.g., *computer* → *mac*
- *generalization*: relax the search constraints, e.g., *lion* → *animal*

Syntactic Relations [Rieh *et al.*, 2006] – Simple to Analyze

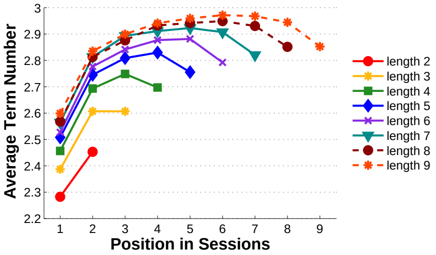
- Syntactic and explicit changes between queries
  - Such as adding terms, removing terms, acronym expansion.
- Clear definitions of reformulation types [Jansen *et al.*, 2009]
- Personalization [Jiang *et al.*, 2011]

Can we exploit user reformulation behavior to QAC?

Number of Terms in Queries

The number of terms will change while adding or removing terms.

- Queries in longer sessions tend to contain more terms.
- The first reformulation increases more than other steps.
- Increase along sessions, and drop near the end of sessions.



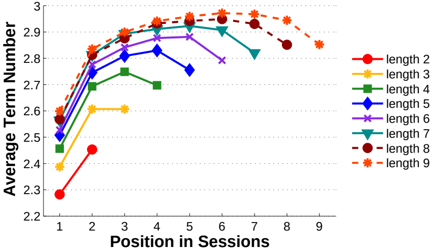
- Helpful to filter intended queries by their lengths syntactically

Do such changes represent some semantic information?

Number of Terms in Queries

The number of terms will change while adding or removing terms.

- Queries in longer sessions tend to contain more terms.
- The first reformulation increases more than other steps.
- Increase along sessions, and drop near the end of sessions.



- Helpful to filter intended queries by their lengths syntactically

Do such changes represent some semantic information?

From Syntactic Relations to Semantic Relations

Semantic Relations

- Specialization: narrow the search constraints
  - More terms are required to describe the intents (constraints).
- Generalization: relax the search constraints
  - Terms (constraints) can be removed.

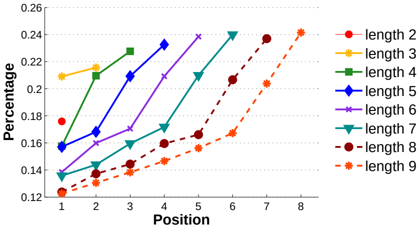
- 2,283 consecutive query pairs from 1,136 sessions are sampled and labeled.
- The syntactic analysis can help us learn semantic relations.

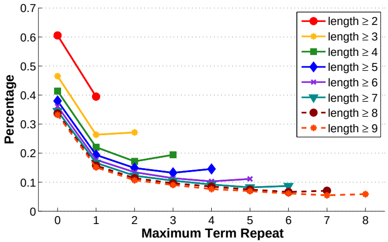
Relation	% in Log	Average Position	Median Position	Change of Term Number	% in Relation	Example
Specialization	27.7%	2.9951	2	Increase	**84.2%**	camera → digital camera
				Decrease	3.7%	perennial plants → stonecrop
				Equal	12.1%	guest book for party → anniversary party guest book
Generalization	12.2%	3.3122	3	Increase	4.0%	airport parking newark → airport parking new york
				Decrease	**82.5%**	great lakes auto → great lakes
				Equal	13.5%	honda blue book → car blue book

Repeated Terms

A usual behavior is to use the repeated terms in previous queries.

"stomach sounds" → *"irritable bowel syndrome"* → *"cramps stomach"*





- Users tend to reuse the terms in the nearest queries.
- Longer sessions are more likely to utilize previously used terms.

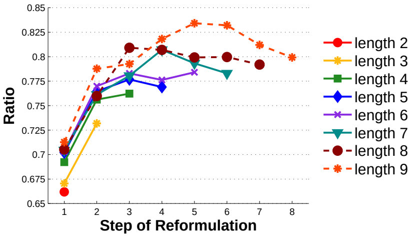
Click Behavior and Repeated Terms

Satisfaction Assumption

The satisfaction (click behavior) might effect a user choose repeated terms.

- 36.06%/**50.54%** of clicking/**no-click** users used repeated terms.
  - If a query is without click, its terms would be reused probably later.

- Difference in the first step of reformulation is the largest.
- The first step is more dependent to click behavior than others.



Summary of User Behavior Analysis

Summary

- The number of terms in queries
  - Trends of syntactic and semantic relations along sessions
- Repeated terms
  - How users utilize terms in the context
- Click behavior and repeated terms
  - How the satisfaction (click behavior) effect users' behavior

Learning user reformulation behavior is helpful for predicting queries!

Query Auto-Completion with Reformulation

Problem Definition

- A session is a sequence of queries $\langle q_1, q_2, \cdots, q_T \rangle$
  - Each query q_i is issued in time t_i, and has c_i clicks.
  - Treat $\langle q_1, q_2, \cdots, q_{T-1} \rangle$ as the context and q_T as the intended query.
- Given the context, the prefix and a candidate set $Q_T = \{q'_j\}$
- The goal is to rank queries in Q_T and let q_T in a high position.

Our Approach

- A supervised framework with *LambdaMART* learning-to-rank model.
- Various reformulation-based features in three categories
  - Term-level, Query-level and Session-level features
  - Attempt to capture how the user changes queries along the session.

Term-level Features

Measure the effectiveness of terms in queries

- Reformulation Types [Akahani *et al.*, 2002]
  - Add terms, Remove terms or Keep terms
  - Encoded as several categorical features
- Term-set Operation
  - Treat a query as a set of terms
  - Union, Intersection, Complacent of context and the query term-sets
  - Estimate how much information conveyed by information need
- Terms contained in both context and the candidate
  - Repeated terms are expected

Query-level Features

Measure relations between context and queries in query-level

- Query Similarity
  - Similar syntactic structures under the same information need
  - term-based cosine similarity and Levenshtein distance are adopted
- Query Length
  - Trend of term numbers
  - Number of terms may not alter rapidly
- Query Frequency
  - Statistical information provided by search logs
  - Relevant frequency to the last query in the context

Session-level Features

Measure how users reformulate queries along whole sessions

- Position Number
  - The stage of the session
  - Reformulation strategies may change over different positions
- Click-through Information
  - Click information is related to term-usage
  - Number of clicks and term set with clicks
- Time Duration (dwell time)
  - Duration of time users stay on the search results
  - Indirectly represent the users' satisfaction

Summary of Reformulation-based Features

Summary

- Term-level features
  - modeled for term effectiveness
  - reformulation types, term-set operation and repeated terms
- Query-level features
  - modeled for query-session relationship
  - query similarity, query length and query frequency
- Session-level features
  - modeled for behavior along whole session
  - position number, click-through information and time duration

Reformulation-based features describe users' behavior in different levels.

Experimental Settings (1/2)

- A commercial search engine log from 1 May, 2013 to 7 May, 2013.
  - Results are consistent and reproducible in public MSN and AOL log.

Data Pre-processing

- 30-minute threshold as the session boundary
- 4-day data for training, the remaining 3-day for testing
- Drop queries appear less than 10 times
- The prefix is the first character of q_T.
- The top-10 frequent queries are the candidate queries.
- Drop sessions with no answers in the candidate set.

Experimental Settings (2/2)

Testing Datasets

- Divide whole testing sessions into four datasets
  - Whole Testing Set (all sessions)
  - Short Sessions (sessions with 2 queries)
  - Medium Sessions (sessions with 3 to 4 queries)
  - Long Sessions (sessions with 5 or more queries)
- Evaluate performance on sessions with different lengths

Evaluation Metrics

- Mean Reciprocal Rank (MRR)
- Success Rate at top-k completions (SR@k)
  - The average percentage of the answers can be found in top-k completions.

- Fine-tune our *LambdaMART* ranking model with parameters of 1,000 decision tress across all experiments.

Four Competitive Baseline Models

- Most Popular Completion (MPC)
  - *Maximum Likelihood Estimation* (MLE) approach
  - Rank candidates by their frequencies
  - The **naïve QAC** baseline approach
- Hybrid Completion (Hyb.C) [Bar-Yossef *et al.*, 2011]
  - **Context-sensitive query completion** method.
  - Consider both context information and the popularity
- Query-based VMM (QVMM) [He *et al.*, 2009]
  - **Context-aware query suggestion** method
  - Learn the probability of query transition along sessions with VMM models
- Concept-based VMM (CACB) [Liao *et al.*, 2011]
  - **Concept-based** context-aware **query suggestion** method
  - Cluster queries into several concepts
  - Learn the concept transition along sessions with VMM models

Overall Performance

Dataset	Measure	MPC	Hyb.C	QVMM	CACB	Our Approach
Whole Testing Set	MRR	0.6415	0.6604 (+2.95%)	0.7137 (+11.25%)	0.7112 (+10.86%)	**0.7433 (+15.87%)**
	SR@1	0.4756	0.5017 (+5.50%)	0.5658 (+18.97%)	0.5593 (+17.61%)	**0.6095 (+28.16%)**
	SR@2	0.6410	0.6625 (+3.36%)	0.7349 (+14.66%)	0.7363 (+14.88%)	**0.7672 (+19.70%)**
	SR@3	0.7623	0.7729 (+1.39%)	0.8293 (+8.79%)	0.8305 (+ 8.94%)	**0.8474 (+11.16%)**
Short Sessions (2 Queries)	MRR	0.6338	0.6335 (−0.04%)	0.7125 (+12.43%)	0.7074 (+11.62%)	**0.7224 (+13.98%)**
	SR@1	0.4654	0.4633 (−0.45%)	0.5636 (+21.10%)	0.5519 (+18.59%)	**0.5794 (+24.49%)**
	SR@2	0.6283	0.6310 (+0.43%)	0.7329 (+16.64%)	0.7348 (+16.95%)	**0.7450 (+18.58%)**
	SR@3	0.7575	0.7567 (−0.10%)	0.8291 (+9.46%)	0.8298 (+9.54%)	**0.8320 (+9.84%)**
Medium Sessions (3 to 4 Queries)	MRR	0.6513	0.6906 (+6.04%)	0.7161 (+9.95%)	0.7160 (+9.93%)	**0.7654 (+17.50%)**
	SR@1	0.4889	0.5443 (+11.33%)	0.5707 (+16.74%)	0.5695 (+16.49%)	**0.6420 (+31.32%)**
	SR@2	0.6552	0.6991 (+6.70%)	0.7369 (+12.47%)	0.7368 (+12.44%)	**0.7892 (+20.45%)**
	SR@3	0.7692	0.7928 (+3.06%)	0.8294 (+7.83%)	0.8305 (+7.98%)	**0.8626 (+12.15%)**
Long Sessions (5 or more Queries)	MRR	0.6522	0.7076 (+8.49%)	0.7130 (+9.32%)	0.7162 (+9.82%)	**0.7842 (+20.24%)**
	SR@1	0.4885	0.5707 (+16.83%)	0.5631 (+15.27%)	0.5676 (+16.20%)	**0.6656 (+36.27%)**
	SR@2	0.6632	0.7149 (+7.79%)	0.7394 (+11.49%)	0.7422 (+11.91%)	**0.8139 (+22.72%)**
	SR@3	0.7674	0.7974 (+3.91%)	0.8300 (+8.16%)	0.8335 (+8.61%)	**0.8798 (+14.65%)**

Overall Performance

Dataset	Measure	MPC	Hyb.C	QVMM	CACB	Our Approach
Whole Testing Set	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Short Sessions (2 Queries)	MRR	0.6338	0.6335 (−0.04%)	0.7125 (+12.43%)	0.7074 (+11.62%)	**0.7224 (+13.98%)**
	SR@1	0.4654	0.4633 (−0.45%)	0.5636 (+21.10%)	0.5519 (+18.59%)	**0.5794 (+24.49%)**
	SR@2	0.6283	0.6310 (+0.43%)	0.7329 (+16.64%)	0.7348 (+16.95%)	**0.7450 (+18.58%)**
	SR@3	0.7575	0.7567 (−0.10%)	0.8291 (+9.46%)	0.8298 (+9.54%)	**0.8320 (+9.84%)**
Medium Sessions (3 to 4 Queries)	MRR	0.6513	0.6906 (+6.04%)	0.7161 (+9.95%)	0.7160 (+9.93%)	**0.7654 (+17.50%)**
	SR@1	0.4889	0.5443 (+11.33%)	0.5707 (+16.74%)	0.5695 (+16.49%)	**0.6420 (+31.32%)**
	SR@2	0.6552	0.6991 (+6.70%)	0.7369 (+12.47%)	0.7368 (+12.44%)	**0.7892 (+20.45%)**
	SR@3	0.7692	0.7928 (+3.06%)	0.8294 (+7.83%)	0.8305 (+7.98%)	**0.8626 (+12.15%)**
Long Sessions (5 or more Queries)	MRR	0.6522	0.7076 (+8.49%)	0.7130 (+9.32%)	0.7162 (+9.82%)	**0.7842 (+20.24%)**
	SR@1	0.4885	0.5707 (+16.83%)	0.5631 (+15.27%)	0.5676 (+16.20%)	**0.6656 (+36.27%)**
	SR@2	0.6632	0.7149 (+7.79%)	0.7394 (+11.49%)	0.7422 (+11.91%)	**0.8139 (+22.72%)**
	SR@3	0.7674	0.7974 (+3.91%)	0.8300 (+8.16%)	0.8335 (+8.61%)	**0.8798 (+14.65%)**

- Hyb.C method is similar to MPC in short sessions.
  - Short sessions have less context.

Overall Performance

- Hyb.C method performs better in longer sessions.
  - Longer sessions have more context.

Dataset	Measure	MPC	Hyb.C	QVMM	CACB	Our Approach
Whole Testing Set	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Short Sessions (2 Queries)	MRR	0.6338	0.6335 (−0.04%)	0.7125 (+12.43%)	0.7074 (+11.62%)	**0.7224 (+13.98%)**
	SR@1	0.4654	0.4633 (−0.45%)	0.5636 (+21.10%)	0.5519 (+18.59%)	**0.5794 (+24.49%)**
	SR@2	0.6283	0.6310 (+0.43%)	0.7329 (+16.64%)	0.7348 (+16.95%)	**0.7450 (+18.58%)**
	SR@3	0.7575	0.7567 (−0.10%)	0.8291 (+9.46%)	0.8298 (+9.54%)	**0.8320 (+9.84%)**
Medium Sessions (3 to 4 Queries)	MRR	0.6513	0.6906 (+6.04%)	0.7161 (+9.95%)	0.7160 (+9.93%)	**0.7654 (+17.50%)**
	SR@1	0.4889	0.5443 (+11.33%)	0.5707 (+16.74%)	0.5695 (+16.49%)	**0.6420 (+31.32%)**
	SR@2	0.6552	0.6991 (+6.70%)	0.7369 (+12.47%)	0.7368 (+12.44%)	**0.7892 (+20.45%)**
	SR@3	0.7692	0.7928 (+3.06%)	0.8294 (+7.83%)	0.8305 (+7.98%)	**0.8626 (+12.15%)**
Long Sessions (5 or more Queries)	MRR	0.6522	0.7076 (+8.49%)	0.7130 (+9.32%)	0.7162 (+9.82%)	**0.7842 (+20.24%)**
	SR@1	0.4885	0.5707 (+16.83%)	0.5631 (+15.27%)	0.5676 (+16.20%)	**0.6656 (+36.27%)**
	SR@2	0.6632	0.7149 (+7.79%)	0.7394 (+11.49%)	0.7422 (+11.91%)	**0.8139 (+22.72%)**
	SR@3	0.7674	0.7974 (+3.91%)	0.8300 (+8.16%)	0.8335 (+8.61%)	**0.8798 (+14.65%)**

Overall Performance

Dataset	Measure	MPC	Hyb.C	QVMM	CACB	Our Approach
Whole Testing Set	MRR	0.6415	0.6604 (+2.95%)	0.7137 (+11.25%)	0.7112 (+10.86%)	**0.7433 (+15.87%)**
	SR@1	0.4756	0.5017 (+5.50%)	0.5658 (+18.97%)	0.5593 (+17.61%)	**0.6095 (+28.16%)**
	SR@2	0.6410	0.6625 (+3.36%)	0.7349 (+14.66%)	0.7363 (+14.88%)	**0.7672 (+19.70%)**
	SR@3	0.7623	0.7729 (+1.39%)	0.8293 (+8.79%)	0.8305 (+ 8.94%)	**0.8474 (+11.16%)**
Short Sessions (2 Queries)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	SR@3	0.7575	0.7567 (−0.10%)	0.8291 (+9.46%)	0.8298 (+9.54%)	**0.8320 (+9.84%)**
Medium Sessions (3 to 4 Queries)	MRR	0.6513	0.6906 (+6.04%)	0.7161 (+9.95%)	0.7160 (+9.93%)	**0.7654 (+17.50%)**
	SR@1	0.4889	0.5443 (+11.33%)	0.5707 (+16.74%)	0.5695 (+16.49%)	**0.6420 (+31.32%)**
	SR@2	0.6552	0.6991 (+6.70%)	0.7369 (+12.47%)	0.7368 (+12.44%)	**0.7892 (+20.45%)**
	SR@3	0.7692	0.7928 (+3.06%)	0.8294 (+7.83%)	0.8305 (+7.98%)	**0.8626 (+12.15%)**
Long Sessions (5 or more Queries)	MRR	0.6522	0.7076 (+8.49%)	0.7130 (+9.32%)	0.7162 (+9.82%)	**0.7842 (+20.24%)**
	SR@1	0.4885	0.5707 (+16.83%)	0.5631 (+15.27%)	0.5676 (+16.20%)	**0.6656 (+36.27%)**
	SR@2	0.6632	0.7149 (+7.79%)	0.7394 (+11.49%)	0.7422 (+11.91%)	**0.8139 (+22.72%)**
	SR@3	0.7674	0.7974 (+3.91%)	0.8300 (+8.16%)	0.8335 (+8.61%)	**0.8798 (+14.65%)**

QVMM outperforms Hyb.C by modeling query transitions.

Overall Performance

Dataset	Measure	MPC	Hyb.C	QVMM	CACB	Our Approach
Whole Testing Set	MRR	0.6415	0.6604 (+2.95%)	0.7137 (+11.25%)	0.7112 (+10.86%)	**0.7433 (+15.87%)**
	SR@1	0.4756	0.5017 (+5.50%)	0.5658 (+18.97%)	0.5593 (+17.61%)	**0.6095 (+28.16%)**
	SR@2	0.6410	0.6625 (+3.36%)	0.7349 (+14.66%)	0.7363 (+14.88%)	**0.7672 (+19.70%)**
	SR@3	0.7623	0.7729 (+1.39%)	0.8293 (+8.79%)	0.8305 (+ 8.94%)	**0.8474 (+11.16%)**
[illegible] (2 Queries)	SR@3	0.7575	0.7567 (−0.10%)	0.8291 (+9.46%)	0.8298 (+9.54%)	**0.8320 (+9.84%)**
Medium Sessions (3 to 4 Queries)	MRR	0.6513	0.6906 (+6.04%)	0.7161 (+9.95%)	0.7160 (+9.93%)	**0.7654 (+17.50%)**
	SR@1	0.4889	0.5443 (+11.33%)	0.5707 (+16.74%)	0.5695 (+16.49%)	**0.6420 (+31.32%)**
	SR@2	0.6552	0.6991 (+6.70%)	0.7369 (+12.47%)	0.7368 (+12.44%)	**0.7892 (+20.45%)**
	SR@3	0.7692	0.7928 (+3.06%)	0.8294 (+7.83%)	0.8305 (+7.98%)	**0.8626 (+12.15%)**
Long Sessions (5 or more Queries)	MRR	0.6522	0.7076 (+8.49%)	0.7130 (+9.32%)	0.7162 (+9.82%)	**0.7842 (+20.24%)**
	SR@1	0.4885	0.5707 (+16.83%)	0.5631 (+15.27%)	0.5676 (+16.20%)	**0.6656 (+36.27%)**
	SR@2	0.6632	0.7149 (+7.79%)	0.7394 (+11.49%)	0.7422 (+11.91%)	**0.8139 (+22.72%)**
	SR@3	0.7674	0.7974 (+3.91%)	0.8300 (+8.16%)	0.8335 (+8.61%)	**0.8798 (+14.65%)**

CACB has no improvement against QVMM because of sparseness.

Overall Performance

Dataset	Measure	MPC	Hyb.C	QVMM	CACB	Our Approach
Whole Testing Set	MRR	0.6415	0.6604 (+2.95%)	0.7137 (+11.25%)	0.7112 (+10.86%)	**0.7433 (+15.87%)**
	SR@1	0.4756	0.5017 (+5.50%)	0.5658 (+18.97%)	0.5593 (+17.61%)	**0.6095 (+28.16%)**
	SR@2	0.6410	0.6625 (+3.36%)	0.7349 (+14.66%)	0.7363 (+14.88%)	**0.7672 (+19.70%)**
	SR@3	0.7623	0.7729 (+1.39%)	0.8293 (+8.79%)	0.8305 (+ 8.94%)	**0.8474 (+11.16%)**
Short Sessions (2 Queries)	MRR	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	SR@1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	SR@2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	SR@3	0.7575	0.7567 (−0.10%)	0.8291 (+9.46%)	0.8298 (+9.54%)	**0.8320 (+9.84%)**
Medium Sessions (3 to 4 Queries)	MRR	0.6513	0.6906 (+6.04%)	0.7161 (+9.95%)	0.7160 (+9.93%)	**0.7654 (+17.50%)**
	SR@1	0.4889	0.5443 (+11.33%)	0.5707 (+16.74%)	0.5695 (+16.49%)	**0.6420 (+31.32%)**
	SR@2	0.6552	0.6991 (+6.70%)	0.7369 (+12.47%)	0.7368 (+12.44%)	**0.7892 (+20.45%)**
	SR@3	0.7692	0.7928 (+3.06%)	0.8294 (+7.83%)	0.8305 (+7.98%)	**0.8626 (+12.15%)**
Long Sessions (5 or more Queries)	MRR	0.6522	0.7076 (+8.49%)	0.7130 (+9.32%)	0.7162 (+9.82%)	**0.7842 (+20.24%)**
	SR@1	0.4885	0.5707 (+16.83%)	0.5631 (+15.27%)	0.5676 (+16.20%)	**0.6656 (+36.27%)**
	SR@2	0.6632	0.7149 (+7.79%)	0.7394 (+11.49%)	0.7422 (+11.91%)	**0.8139 (+22.72%)**
	SR@3	0.7674	0.7974 (+3.91%)	0.8300 (+8.16%)	0.8335 (+8.61%)	**0.8798 (+14.65%)**

Our approach significantly outperforms all baselines.

Overall Performance

Dataset	Measure	MPC	Hyb.C	QVMM	CACB	Our Approach
Whole	MRR	0.6415	0.6604 (+2.95%)	0.7137 (+11.25%)	0.7112 (+10.86%)	**0.7433 (+15.87%)**
	SR@1	0.4756	0.5017 (+5.50%)	0.5658 (+18.97%)	0.5593 (+17.61%)	**0.6095 (+28.16%)**
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	**0.7672 (+19.70%)**
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	**0.8474 (+11.16%)**
Short Sessions (2 Queries)	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	**0.7224 (+13.98%)**
	SR@1	0.4654	0.4633 (−0.45%)	0.5636 (+21.10%)	0.5519 (+18.59%)	**0.5794 (+24.49%)**
	SR@2	0.6283	0.6310 (+0.43%)	0.7329 (+16.64%)	0.7348 (+16.95%)	**0.7450 (+18.58%)**
	SR@3	0.7575	0.7567 (−0.10%)	0.8291 (+9.46%)	0.8298 (+9.54%)	**0.8320 (+9.84%)**
Medium Sessions (3 to 4 Queries)	MRR	0.6513	0.6906 (+6.04%)	0.7161 (+9.95%)	0.7160 (+9.93%)	**0.7654 (+17.50%)**
	SR@1	0.4889	0.5443 (+11.33%)	0.5707 (+16.74%)	0.5695 (+16.49%)	**0.6420 (+31.32%)**
	SR@2	0.6552	0.6991 (+6.70%)	0.7369 (+12.47%)	0.7368 (+12.44%)	**0.7892 (+20.45%)**
	SR@3	0.7692	0.7928 (+3.06%)	0.8294 (+7.83%)	0.8305 (+7.98%)	**0.8626 (+12.15%)**
Long Sessions (5 or more Queries)	MRR	0.6522	0.7076 (+8.49%)	0.7130 (+9.32%)	0.7162 (+9.82%)	**0.7842 (+20.24%)**
	SR@1	0.4885	0.5707 (+16.83%)	0.5631 (+15.27%)	0.5676 (+16.20%)	**0.6656 (+36.27%)**
	SR@2	0.6632	0.7149 (+7.79%)	0.7394 (+11.49%)	0.7422 (+11.91%)	**0.8139 (+22.72%)**
	SR@3	0.7674	0.7974 (+3.91%)	0.8300 (+8.16%)	0.8335 (+8.61%)	**0.8798 (+14.65%)**

- Performs better in longer sessions
  - Longer sessions are easier to model behavior

Summary of Overall Performance

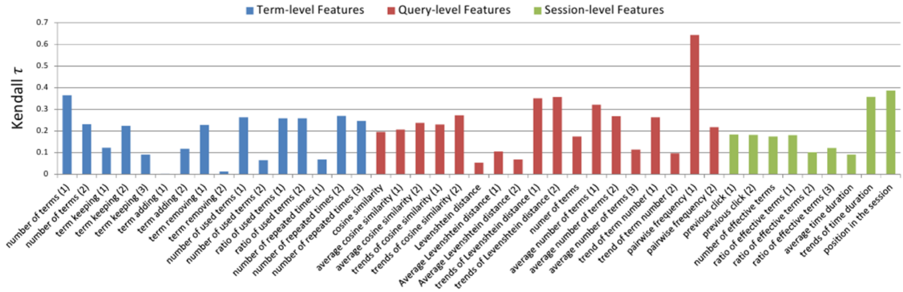
For baseline approaches

- Hyb.C method is similar to MPC in short sessions (less context)
- Hyb.C method performs better in longer sessions (more context)
- QVMM outperforms Hyb.C by modeling query transitions
- CACB has no improvement against QVMM because of sparseness

For our approach

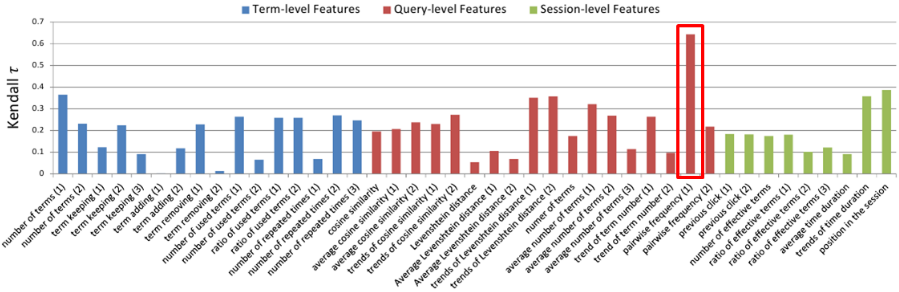
- Significantly outperforms all of baseline approaches
- Performs better in longer sessions (easier to model behavior)

Feature Effectiveness Analysis



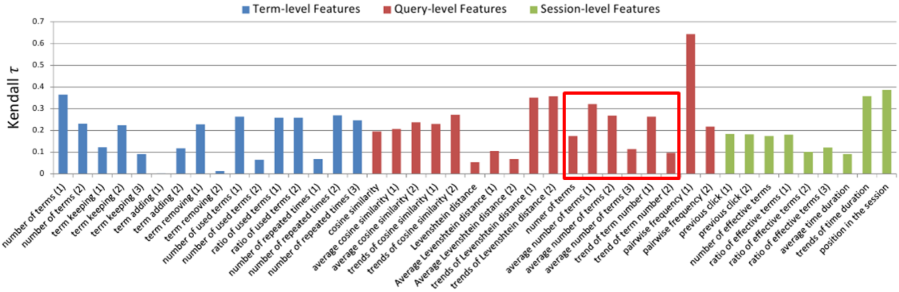
- The query-frequency is the most significant feature (conventional approaches)
- Query length is useful (the analysis of term numbers)
- Most of term-level features are helpful (modeling complex reformulation behavior)
- The position in the session is highly related (reformulation stage)
- Clicks (satisfaction) and time duration are also effective.

Feature Effectiveness Analysis



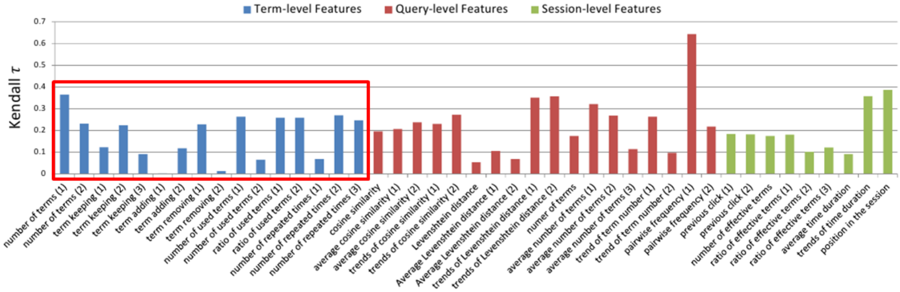
- The query-frequency is the most significant feature (conventional approaches)
- Query length is useful (the analysis of term numbers)
- Most of term-level features are helpful (modeling complex reformulation behavior)
- The position in the session is highly related (reformulation stage)
- Clicks (satisfaction) and time duration are also effective.

Feature Effectiveness Analysis



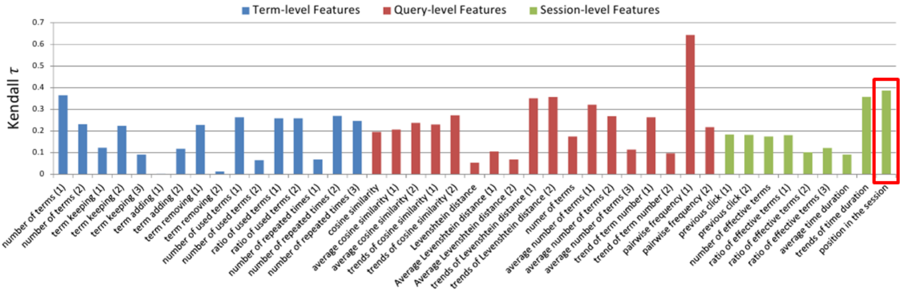
- The query-frequency is the most significant feature (conventional approaches)
- Query length is useful (the analysis of term numbers)
- Most of term-level features are helpful (modeling complex reformulation behavior)
- The position in the session is highly related (reformulation stage)
- Clicks (satisfaction) and time duration are also effective.

Feature Effectiveness Analysis



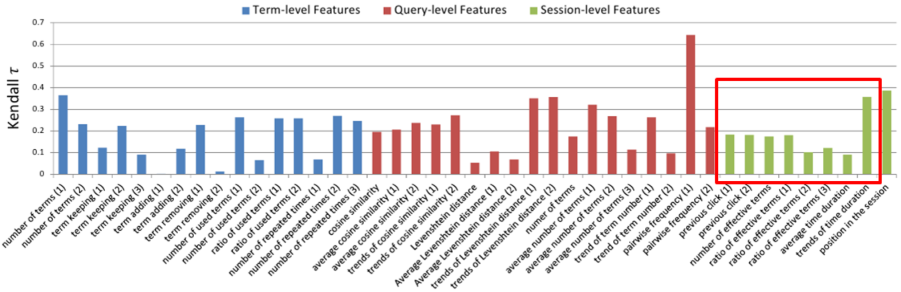
- The query-frequency is the most significant feature (conventional approaches)
- Query length is useful (the analysis of term numbers)
- Most of term-level features are helpful (modeling complex reformulation behavior)
- The position in the session is highly related (reformulation stage)
- Clicks (satisfaction) and time duration are also effective.

Feature Effectiveness Analysis



- The query-frequency is the most significant feature (conventional approaches)
- Query length is useful (the analysis of term numbers)
- Most of term-level features are helpful (modeling complex reformulation behavior)
- The position in the session is highly related (reformulation stage)
- Clicks (satisfaction) and time duration are also effective.

Feature Effectiveness Analysis



- The query-frequency is the most significant feature (conventional approaches)
- Query length is useful (the analysis of term numbers)
- Most of term-level features are helpful (modeling complex reformulation behavior)
- The position in the session is highly related (reformulation stage)
- Clicks (satisfaction) and time duration are also effective.

Application: Query Suggestion

- Query suggestion is an application of our approach.
- Queries in high positions may be also relevant.
- The adjacency frequency $P(q_T|q_{T-1})$ is the naïve baseline.
- Experimental settings
  - Sample 100 sessions from testing data and apply 3 approaches
  - Manually labeling top 15 queries and evaluate with NDCG

NDCG	Adj. Freq.	QVMM	Our Approach
@5	0.5817	**0.6036 (+3.76%)**	0.5973 (+2.68%)
@10	0.5941	0.6152 (+3.55%)	**0.6175 (+3.94%)**
@15	0.6949	0.7090 (+2.03%)	**0.7127 (+2.56%)**

Conclusions

- Extensive analysis shows reformulation behavior is helpful for QAC
- Propose a supervised approach for query auto-completion
- Our approach requires less data for training
- Our approach considers different user behavior for reformulation
- All of three-type features are useful and important.
- Our approach actually helps users save their keystrokes.

Thank you for listening! Questions?

Conclusions

- Extensive analysis shows reformulation behavior is helpful for QAC
- Propose a supervised approach for query auto-completion
- Our approach requires less data for training
- Our approach considers different user behavior for reformulation
- All of three-type features are useful and important.
- Our approach actually helps users save their keystrokes.

Thank you for listening! Questions?