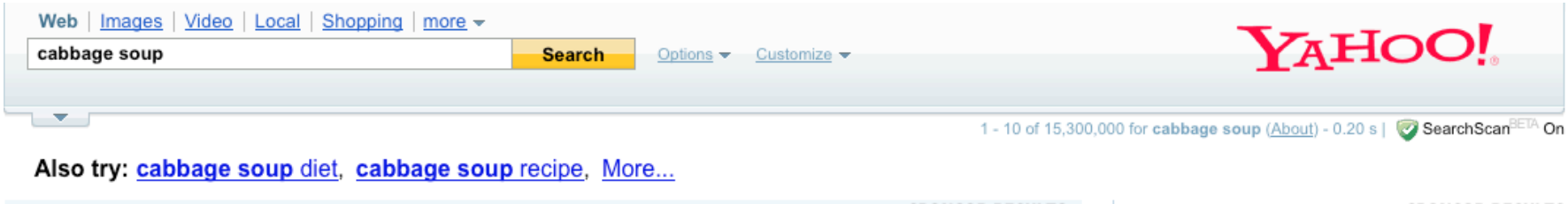
Online Expansion of Rare Queries for Sponsored Search

Peter Ciccolo, Evgeniy Gabrilovich, Vanja Josifovski, Don Metzler, Lance Riedel, Jeff Yuan

Yahoo! Research

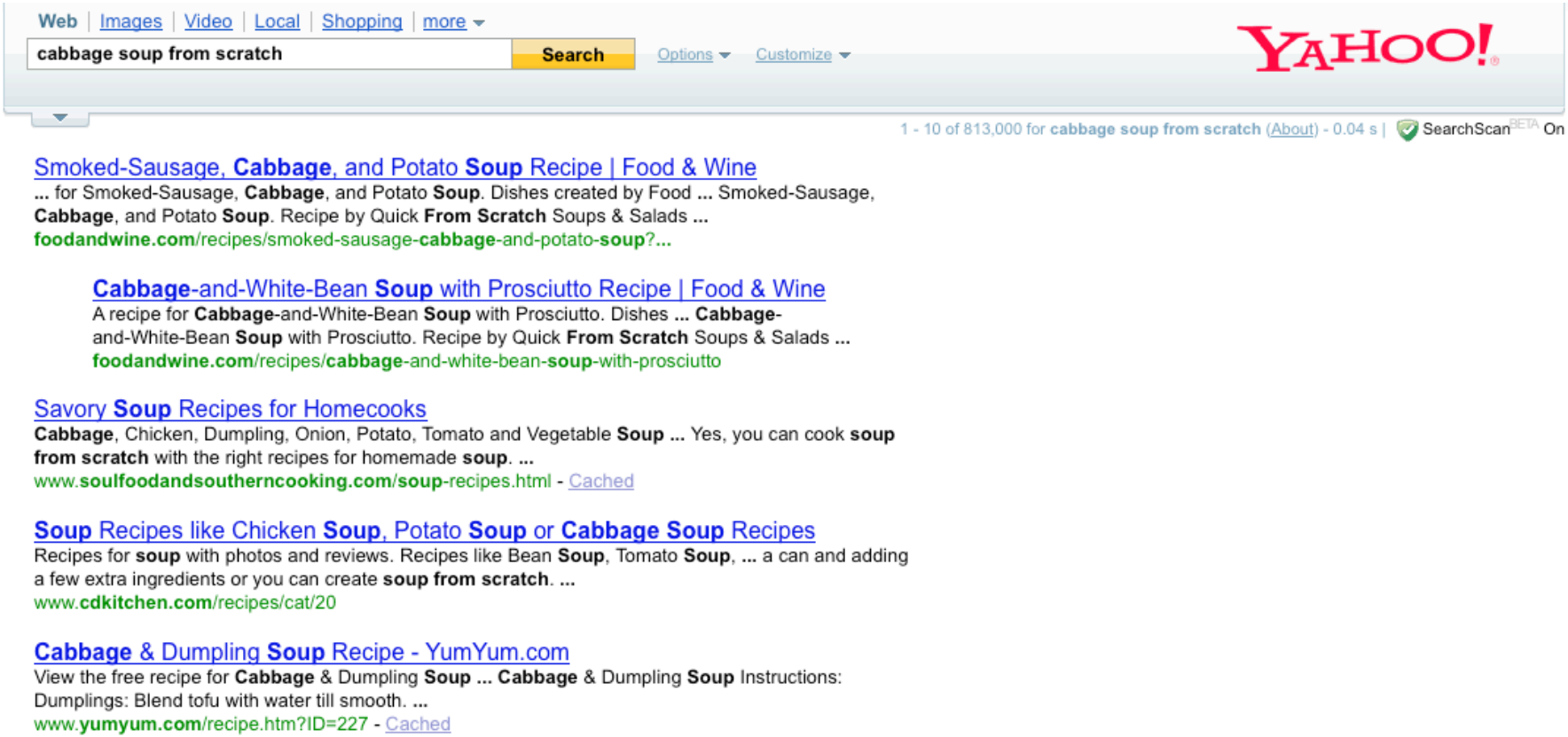
Sponsored Search



Sponsored Search in the Tail

- Many tail queries display no sponsored search results
- Advertising in the tail is challenging
  - Longer / rare queries are more difficult to interpret
  - Exact match and phrase match are much less likely
  - Click-based relevance predictors are poor, due to data sparseness
- Considerable monetization potential in tail, if done correctly

Sponsored Search in the Tail



Advanced Match Scoring

- Advanced match (or broad match) refers to an inexact match between the query and an ad's bid phrase
  - Advertisers opt in
  - Tend to have lower click-through rates
- Matching queries to bid phrases is challenging
  - Sparse representation (just a few terms)
  - Vocabulary mismatch
  - Misspellings
- Query and/or bid phrase expansion is often used to overcome these issues

Offline Query Expansion

- Query expansion is generally expensive to perform in real time
- It is possible to compute expansions for popular queries offline and cache the results
- Query expansion sources
  - Query reformulations (query logs, co-clicks on search results and ads, etc.)
  - Top features from web search results (next slide)
- Circumstantial evidence can convince better than one witness!
- Expanded query is a combination of the original query and additional information from expansion sources (more details later)

Query Expansion using Search Results

- Query expansion using web search results has been shown to be effective for sponsored search in the past [Broder et al., SIGIR ’07, Radlisnki et al., SIGIR ‘08]

- General overview:
  - Run query against web search engine
  - Retrieve the top *N* pages (we use 40)
  - Extract features and assign weights (*TF*.*IDF*)
  - Select (about 100) features by sorting features by weight
  - End result is a query -> feature vector

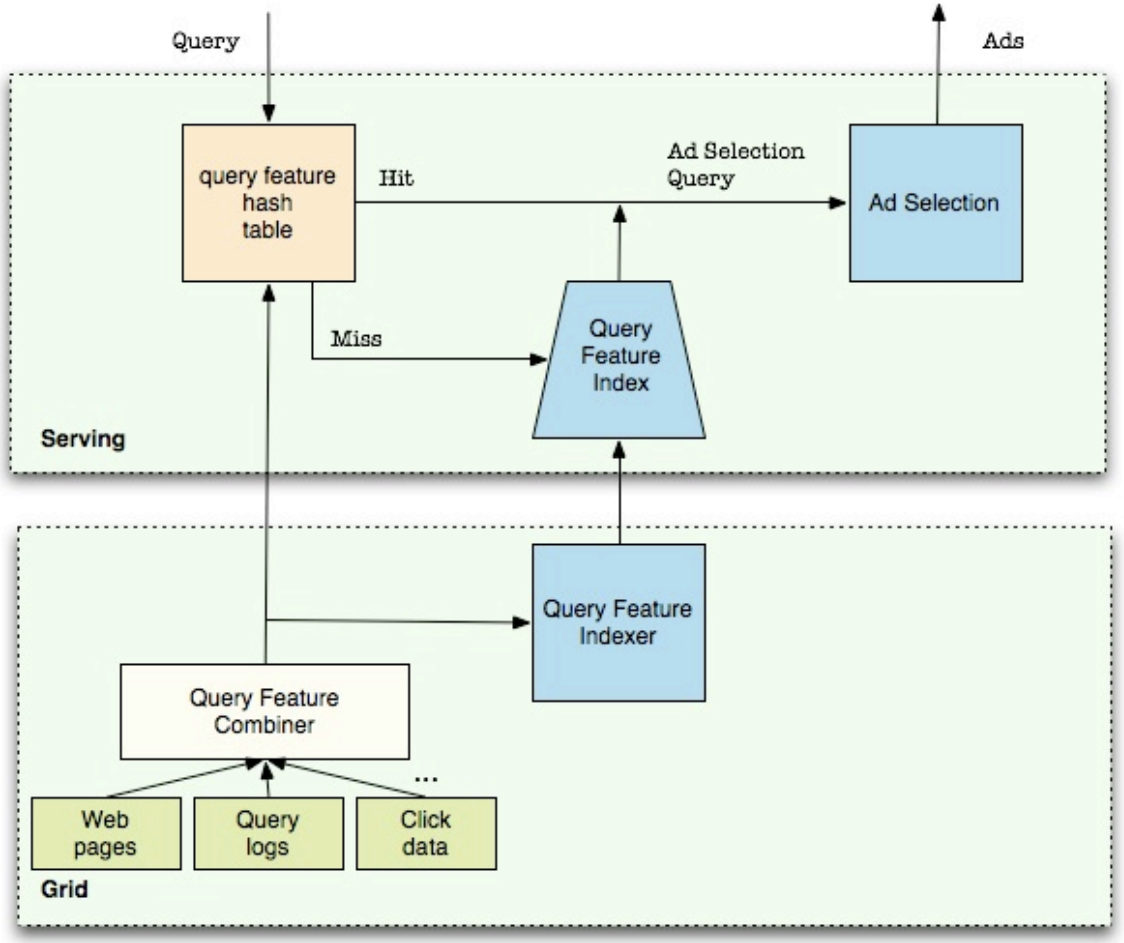
- Helps provide uniformity of the search engine result page!

What about the Tail?

- Expansion is effective, but only if the incoming query is in the expansion cache
- Can we leverage the offline expansion cache to improve ad matching for queries that are not in the cache (rare queries)?

  … and do it in real time?
- Our proposed methodology
  - ‘Map’ incoming rare query to related popular queries
  - Expand using offline expansion cache
  - Use expanded version of rare query for ad matching

System Architecture



Online Query Expansion

- Map rare queries to head/torso queries that have been processed offline via similarity search
- Build inverted index of queries that have been pre-expanded
  - Document retrieval analogy: query ≈ docid, expansion vector ≈ document text
  - Queries run against index will match against the expanded versions of the pre-expanded queries
- If an incoming query was not pre-processed, it is run against the inverted index of pre-expanded query vectors
    - Retrieve *k* most related head/torso queries
    - Construct expanded query vector
    - This query can be executed very efficiently, adding little overhead

Online Query Expansion Example

cabbage soup from scratch

feature	inverted list
cabbage	cabbage (10), cabbage soup (8), cabbage soup diet (5), leafy vegetable (3), vegetable (2)
soup	soup (10), homemade soup (8), recipes (7), tomato soup (5), cabbage soup (2), food (2)
“cabbage soup”	cabbage soup (10), cabbage soup diet (8), cabbage (7), diet (5), weight loss (3)
scratch	scratch (10), scratchy eyes (3), cat scratch (3)
“from scratch”	recipes (10), cooking (7), food (5)

Online Query Expansion Example

cabbage soup from scratch

↓

feature	inverted list
cabbage	cabbage (10), cabbage soup (8), cabbage soup diet (5), leafy vegetable (3), vegetable (2)
soup	soup (10), homemade soup (8), recipes (7), tomato soup (5), cabbage soup (2), food (2)
“cabbage soup”	cabbage soup (10), cabbage soup diet (8), cabbage (7), diet (5), weight loss (3)
scratch	scratch (10), scratchy eyes (3), cat scratch (3)
“from scratch”	recipes (10), cooking (7), food (5)

↓

1. cabbage soup
2. cabbage soup diet
3. recipes
4. food

Online Query Expansion Example

cabbage soup from scratch

feature	inverted list
cabbage	cabbage (10), cabbage soup (8), cabbage soup diet (5), leafy vegetable (3), vegetable (2)
soup	soup (10), homemade soup (8), recipes (7), tomato soup (5), cabbage soup (2), food (2)
“cabbage soup”	cabbage soup (10), cabbage soup diet (8), cabbage (7), diet (5), weight loss (3)
scratch	scratch (10), scratchy eyes (3), cat scratch (3)
“from scratch”	recipes (10), cooking (7), food (5)

1. cabbage soup
2. cabbage soup diet
3. recipes
4. food

Used to expand original query

Query Feature Weighting

Original query weights:

$$w(f,Q) = (1 + \log \#(f,Q)) \cdot idf(f)$$

Offline expansion query feature weights:

$$w(f,E(Q)) = \left(1 + \log \sum_{D \in Results(Q)} \#(f,D)\right) \cdot idf(f)$$

Similarity between the query vectors:

$$sim(Q,E(Q')) = \sum_{F \in \{F_u,F_p,F_c\}} \lambda_F \cdot sim_F(Q,E(Q'))$$

Final query weights:

$$w(f,Q^*) = (1-\lambda) \cdot w(f,Q) + \lambda \sum_{Q' \in Related(Q)} \frac{w(f,Q')}{|Related(Q)|}$$

Ad Feature Weighting and Scoring

- Ad features are weighted using BM25F

$$w(f,A) = \frac{(k+1)\cdot \#(f,A)}{k\cdot\left((1-b)+b\cdot\frac{|A|}{|A|_{avg}}\right)+\#(f,A)}\cdot idf(f)$$

  - #(f, A) is a field weighted term frequency

- Final ad scoring function:

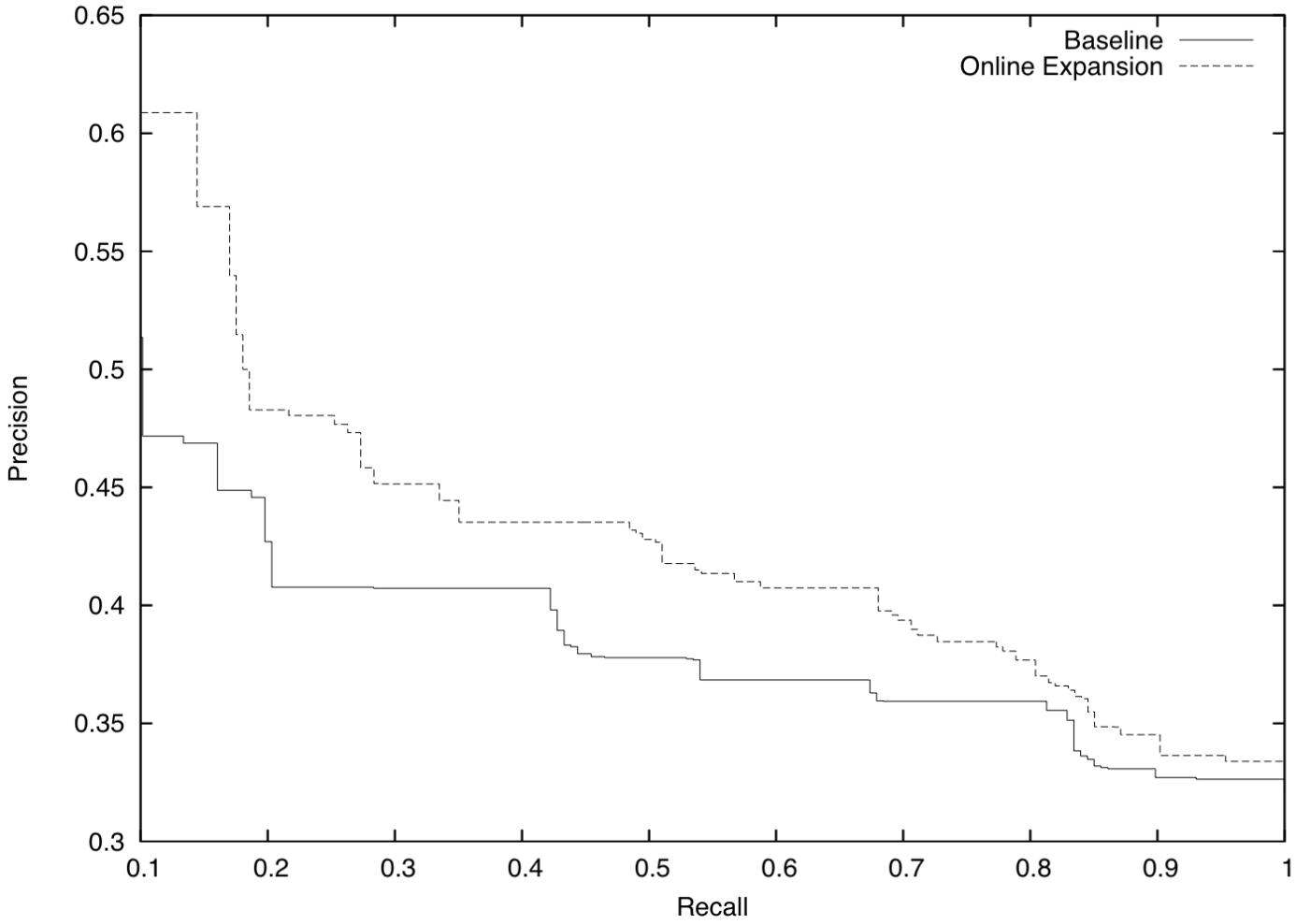
$$S(Q,Q^*,A)=\sum_{F\in\{F_u,F_p,F_c\}}\lambda_F\cdot sim_F(Q^*,A)\cdot(1+prox_F(Q,A))$$

  - sim(Q*, A) is the cosine similarity between the expanded query vector and the ad vector
  - prox(Q,A) is a simple proximity measure between original query and the ad

Experimental Evaluation

- Expansion cache consists of a large set of pre-expanded queries
  - Most frequent Yahoo! web search queries
  - Bid phrases from Yahoo!'s textual ad corpus
- Randomly sampled 400 rare queries from a web search query log
  - 121 in lookup table
  - 179 not in lookup table
- Human editors judged top 3 results for several system variants
  - Total of 3,556 judgments
  - Judgments are on an integral scale 1-5 (1=highly attractive, 5=poor)
- Evaluate using interpolated precision-recall curves and DCG

Queries Not Found in Lookup Table

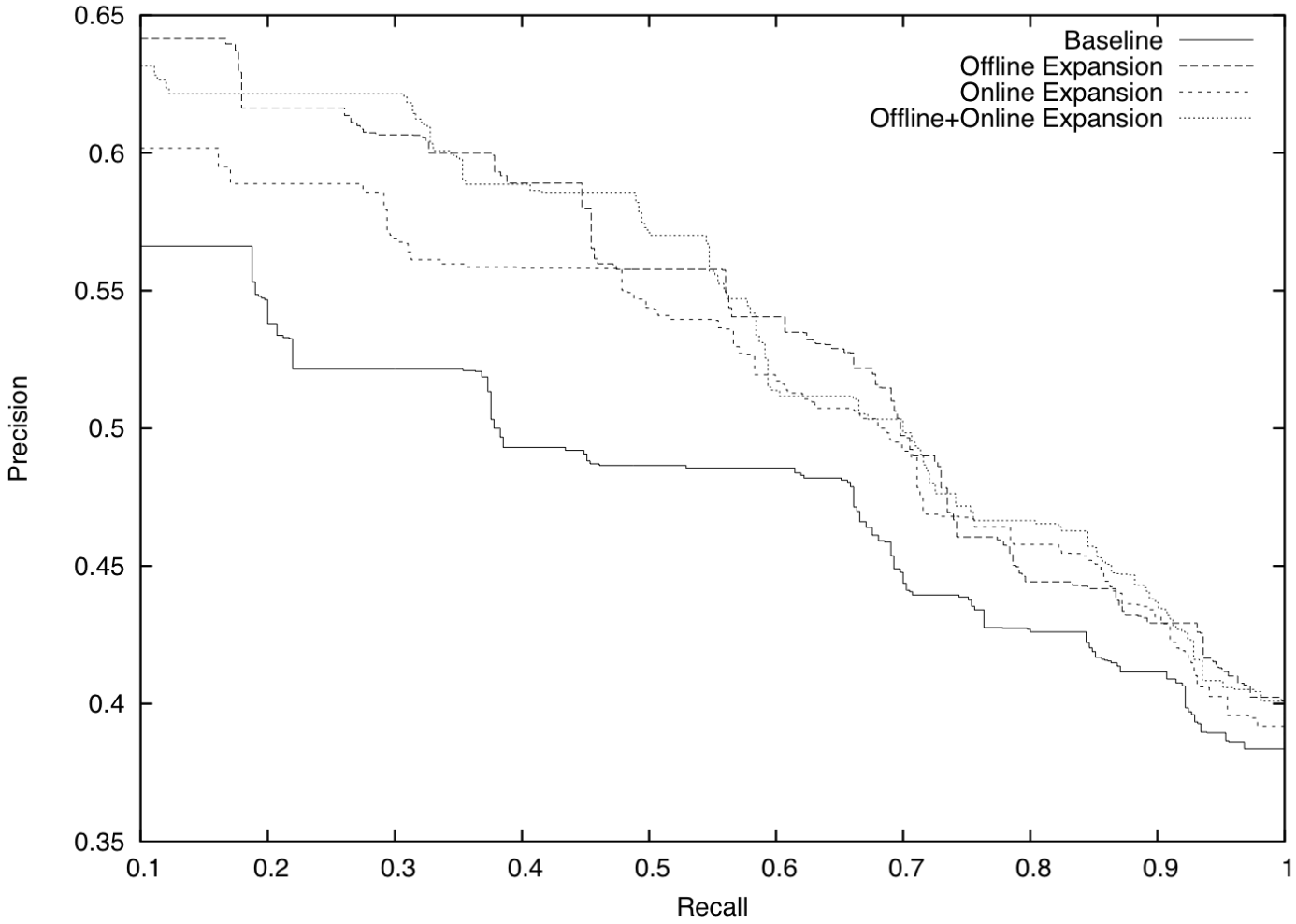


Queries Not Found in Lookup Table

	Baseline	Online Expansion
DCG@1	0.99	1.07 (+8.1%)†
DCG@2	1.57	1.66 (+5.7%)†
DCG@3	1.97	2.10 (+6.6%)‡

Summary: Our proposed online expansion approach significantly improves DCG over the baseline approach

Queries Found in Lookup Table



Queries Found in Lookup Table

	DCG@1	DCG@2	DCG@3
Baseline	2.89	4.56	5.75
Online Expansion	2.83	4.43	5.54
Offline Expansion	3.07‡	4.75	5.87
Online+Offline Expansion	2.91	4.44	5.59

Summary: Online expansion less useful for queries found in lookup table. Offline expansion tends to be more effective for these queries.

Hybrid Online / Offline Approach

- Results so far…
  - Offline expansion works best for queries in lookup table
  - Online expansion works best for queries not in lookup table
- Rather than blindly applying one approach to all rare queries, we can use an adaptive approach that gets the best of both worlds
  - Use offline expansion for queries in lookup table
  - Use online expansion for queries not in lookup table
- We refer to this as the “hybrid” approach

All Rare Queries (in Table + not in Table)

	DCG@1	DCG@2	DCG@3
Baseline	1.61	2.58	3.23
Online Expansion	1.71‡	2.68	3.32
Offline Expansion	1.66	2.63	3.25
Online+Offline Expansion	1.76‡	2.69	3.37
Hybrid	1.79†	2.78†	3.43†

Summary: Online expansion is necessary to achieve good results for rare queries. The hybrid approach is superior to any other single method.

Conclusions

- Producing relevant advanced match ads for tail queries is challenging
- We proposed a method that leverages pre-processed head/torso query expansions to improve rare/tail queries
- Tail queries are mapped to head/torso queries, via similarity search against an inverted index of offline expansions
  - Can (theoretically) expand 100% of the query volume
  - Nominal latency overhead
- Experimental results show significant improvements in DCG (+8%) and precision at all recall levels

Thank you!