IN THE NAME OF GOD

Synthesizing High Utility Suggestions for Rare Web Search Queries

Learning to Suggest: A Machine Learning Framework for Ranking Query Suggestions

Query Suggestion

- Goal : Assist the user to get what she needs.
- Other Problems:
  - Query Completion
  - Spell Correction
  - Query rewriting
- We are talking about post-submit query reformulation

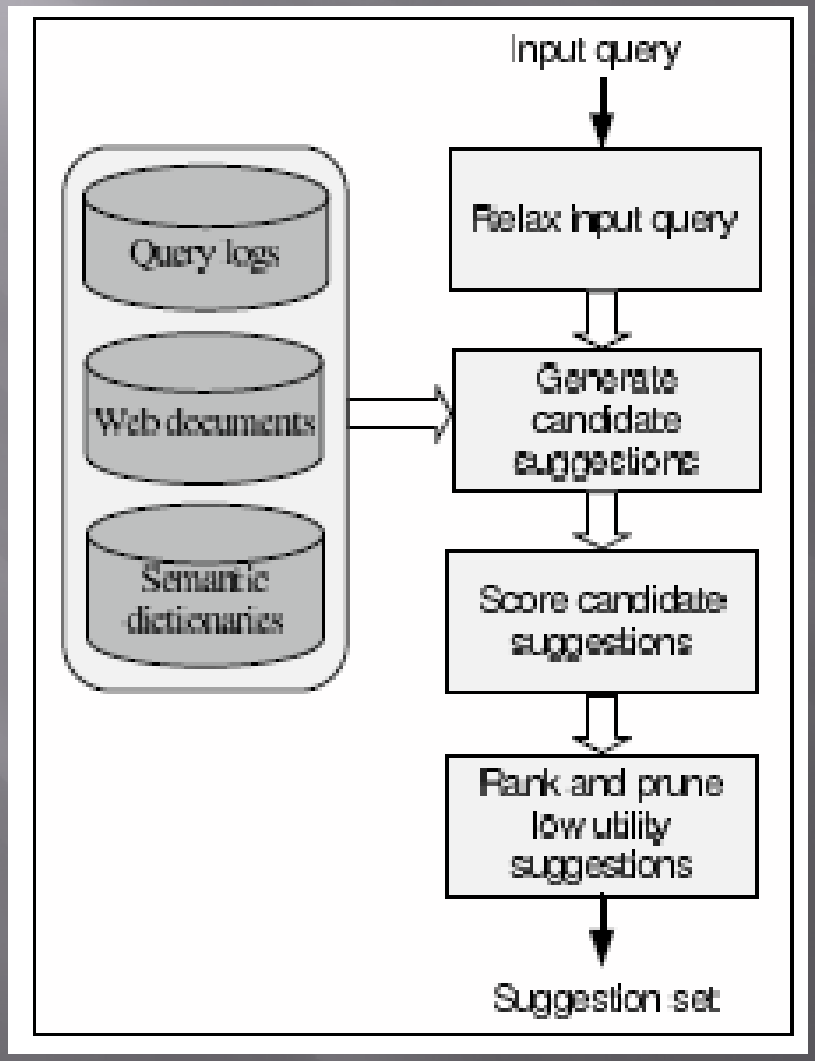
Methods

- Search logs
  - wisdom of crowds
- Synthetic suggestions
  - Specialization
  - Generalization
  - Laterals
  - Partial lateral

These Papers

- Method
  - Synthetic Suggestions
- Main gain
  - better coverage
- Simple Ideas yet effective

Query Suggestion Steps



Synthetic suggestion challenges

- Identifying important concepts in a query
- Identifying candidates for tranformation
  - Supercuts new jersy
- Well-formedness of suggestions
  - Tampa bay fisheries -> Tampa tribune fisheries

Generating Suggestion Candidates

- Co-occurrence in query sessions
  - Only for frequent queries
- Semantic relations from web corpus
  - Similar distribution -> synonym
- Co-clicked URLs
- Context from original query
  - Push back dropped words

Query Relaxation

- Itself a candidate
- Deals with log info sparseness
  - big lots *furniture* ***store***
- Critical and noncritical terms are learned from how users reformulate their own queries

Co-clicked URLs

- Based on a substitution dictionary
  - Could be in a context-aware manner
- Query-URL bipartite graph
  - Remove URLs connected to more than 200 queries
  - Sometimes replace URL by its domain

Ranking Suggestions

- Well-formedness
  - Language models
- Relevance
  - Click over similarity
  - Context vector similarity
  - Web based aboutness
- Utility
  - More results are needed

Web based aboutness

- Looks at the results set
- Is not based on the query logs
  - Deals with sparsity
- Full coverage

Aboutness vector

- **Require** : Concept dictionary D, query q

- 1. Retrieve set R of top-k results for q
- 2. T = Terms from D contained in R
- 3. Eliminate from T terms in q
- 4. for each term t in D do
- 5. d(t) = number of results that t appears
- 6. r(t) = total rank that t appears
- 7. R(t) = (((k+1)d(t)) – r(t)) / d(t)k
- 8. S(t) = d(t)R(t) / k
- 9. end for
- 10. Get the 20 terms with highest score S(t)

- **Ab(q)**=[S(t1),….,S(t20)]

Suggestion Utility

$$p(e(u_{si}) = 1|q_s) = d(u_{si}, q_s) = \frac{1}{\log_2(r_i + 1)}$$

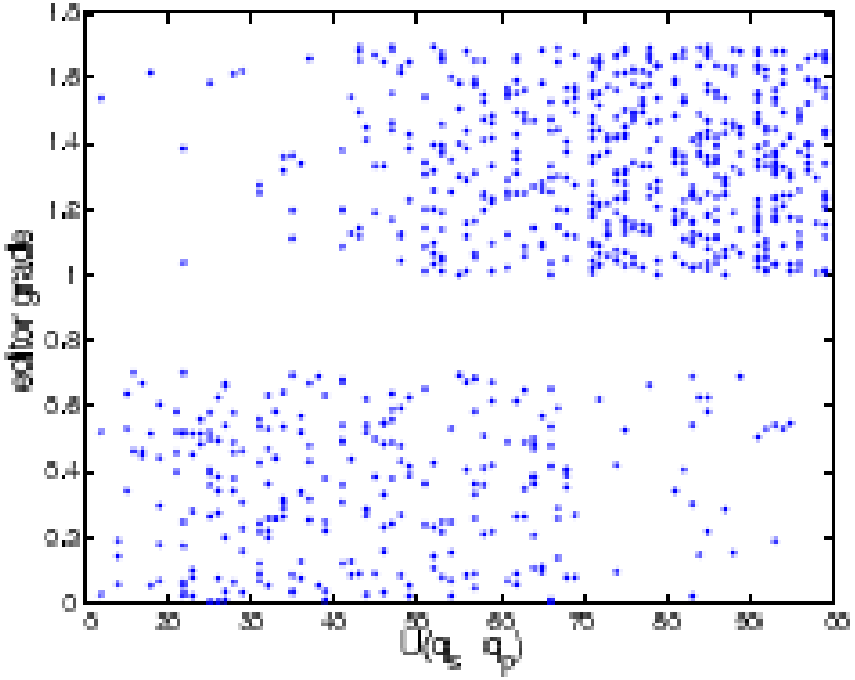
$$p(e(u_{si}) = 1\ q_p) =$$

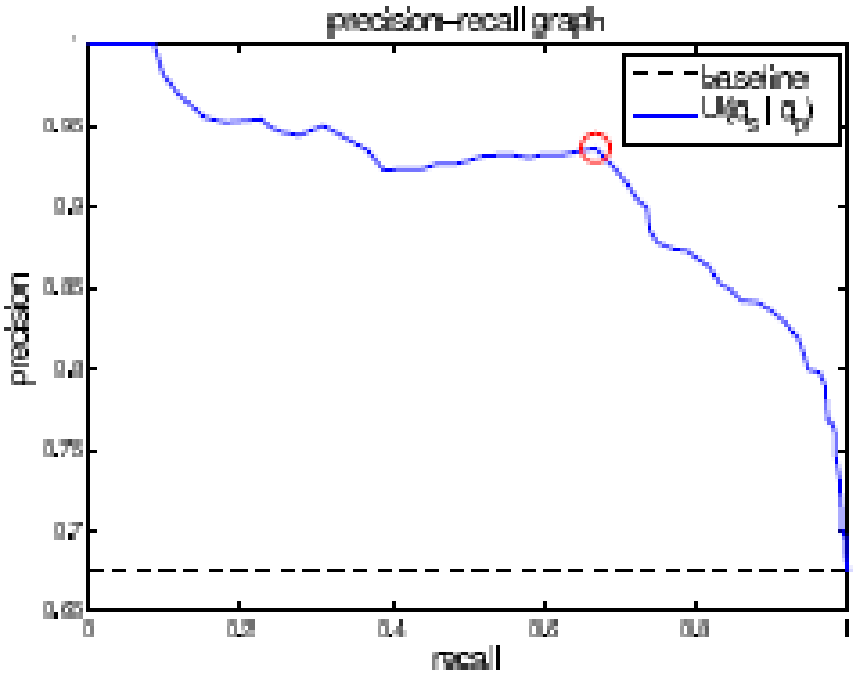
$$\begin{cases} 0 & : u_{si} \notin URL_p \\ 1 & : \begin{array}{c} u_{si} \in URL_p, \\ E\{d(q_p, u_{si})\} \geq E\{d(q_s, u_{si})\} \end{array} \\ \frac{E\{d(q_p, u_{si})\}}{E\{d(q_s, u_{si})\}} & : \begin{array}{c} u_{si} \in URL_p, \\ E\{d(q_p, u_{si})\} < E\{d(q_s, u_{si})\} \end{array} \end{cases}$$

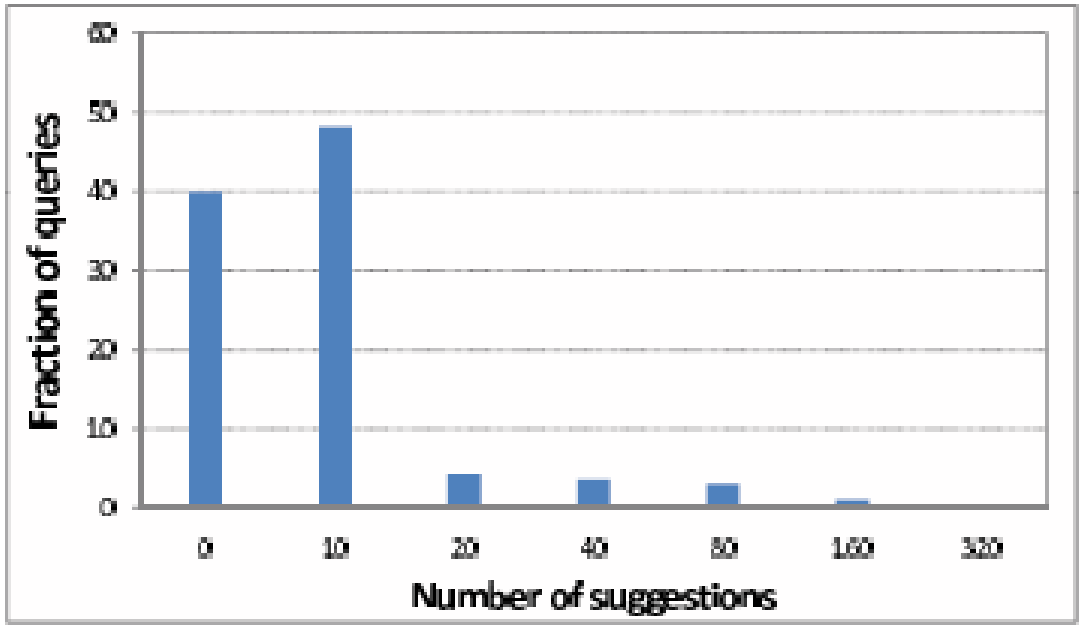
$$U(q_s|q_p) = 1 - \sum_{u \subset URL_s} p(c(u) = 1|q_s)\, p(e(u) = 1|q_p)$$

- The suggestion candidates are pruned with a threshold of γ

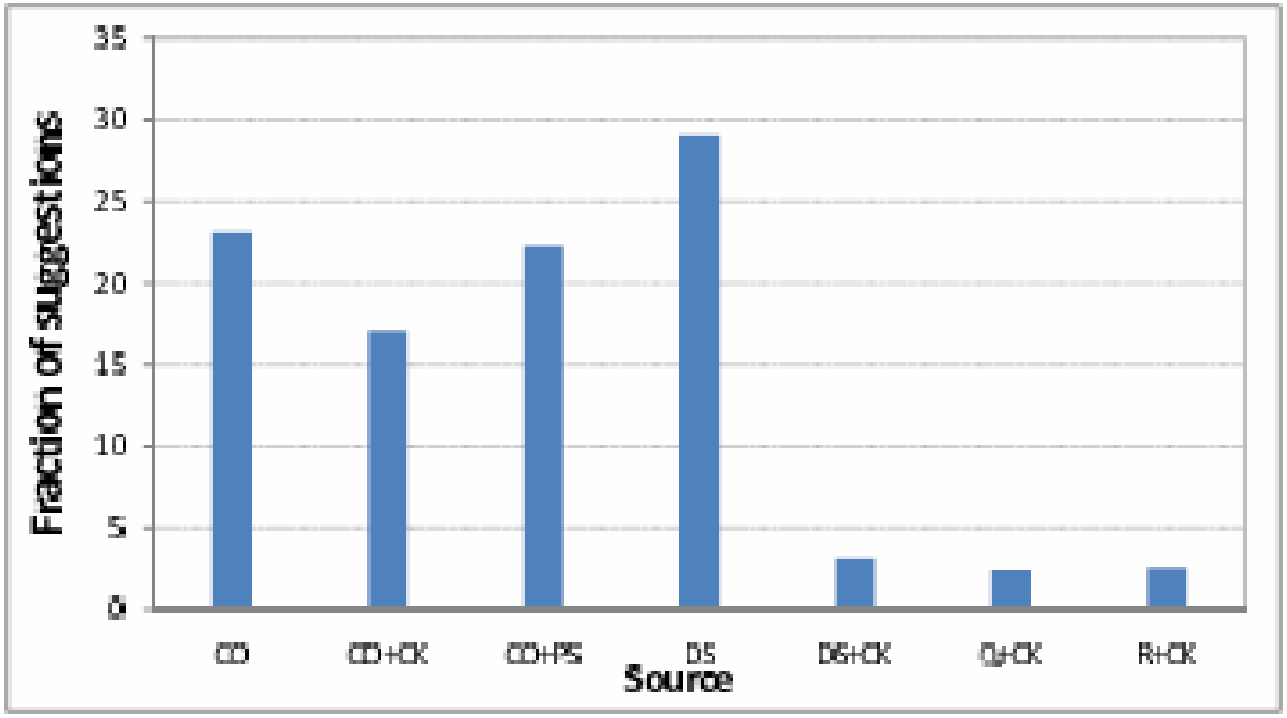
y estimation







Main gain: Coverage



Learning Paper

- The same problem and methods in a ML environment
- The Learning algorithm is GBDT
  - Hyper parameters are tuned by cross validation
    - Number of trees
    - Number of nodes
    - Shrinkage factor

Learning Paper(Cont.)

- Feature Set

Table 2: Lexical features

Feature	Short Description
LEV	Levenshtein distance between q_1 and q_2
LQ1	length of q_1
LQ2	length of q_2
LDIFF	LQ1-LQ2
ABSLDIFF	abs(LDIFF)
ABSLDIFFN	ABSLDIFF/LQ1
NW1	number of tokens in q_1
NW2	number of tokens in q_2
COMMONW	number of tokens in common in q_1 and q_2
COMMONWN	COMMONW/NW1
COMMONWP	tokens in common in the beginning
COMMONWS	tokens in common at the end
COMMONCP	characters in common in the beginning
COMMONCS	characters in common at the end

Table 3: Result set features

Feature	Short Description
LTR11	LTR score of the top result for q_1
LTR21	LTR score of the top result for q_2
LTR15	average LTR score at top 5 for q_1
LTR25	average LTR score at top 5 for q_2
LTR110	average LTR score at top 10 for q_1
LTR210	average LTR score at top 10 for q_2
LTRRATIO1	LTR11/LTR21
LTRDIFF1	LTR11-LTR21
LTRRATIO5	LTR15/LTR25
LTRDIFF5	LTR15-LTR25
LTRRATIO10	LTR110/LTR210
LTRDIFF10	LTR110-LTR210
COURLCOUNT1	boolean, URLs in top result are the same
COURLCOUNT5	number of the same URLs in top 5
COURLCOUNT10	number of the same URLs in top 10
CODOMAINCOUNT1	boolean, domains of the top results are the same
CODOMAINCOUNT5	number of same domains in top 5
CODOMAINCOUNT10	number of same domains in top 10
CODOMAINIQF1	IQF of of the domain at top result, if the same
CODOMAINIQF5	total IQF of the common domains in top 5
CODOMAINIQF10	total IQF of the common domains in top 10
ABOUTNESS	cosine similarity over the aboutness vectors

$$\mathrm{IQF}(d) = \log\left(\frac{|Q|}{|q : d \in q|}\right)$$

Aboutness vector computing algorithm

- **Require**: Concept dictionary D, query q.

- 1: Retrieve set R of top-k results for q.
- 2: for concept t 2 D do
- 3: S(t) = 0
- 4: for i = 1, .., k do
- 5: d = i-th document in R.
- 6: if concept t is in d then
- 7: a = aboutness score of concept t in D [19].
- 8: $S(t) = S(t) + 0.9^{i-1}a$.
- 9: end if
- 10: end for
- 11: end for
- 12: Set S(t) to 0 for the concepts t that are not in the top 20
- highest scores.

- 13: **Aboutness vector**: a(q) := [. . . , S(t), . . .]

Learning Paper (Cont.)

- Implicit Task Boundary Detection
  - Idea: Pr(q2) has a very large entropy, but the entropy of Pr(q2| q1, c = 1) should be much smaller
- Predicting Utility
  - Addressed no click problem

    (no click + no further reformulation + answer module)
- Two more suggestion Ideas:
  - Common terms in queries leading to clicks to the same URLs
  - Most Frequent Extensions
    - Similar to the query completion problem

Learning Paper (Cont.)

- Two design decisions on co-occurrence
  - Taking only consecutive pairs in a session into account
    - Because of Intention drifting
  - Taking all query pairs in a session into account
    - Since Intention drifting is not so common

Learning Paper (Cont.)

	depth = 1	depth=3	depth=5	depth=7	depth=9	depth=12
Coverage						
10M	-9%	2%	2%	4%	34%	294%
10MC	-9%	0%	0%	2%	33%	282%
10M-ML	-13%	-3%	-1%	2%	32%	289%
10MC-ML	-17%	-7%	-6%	-4%	25%	269%
10M-ML-SY	17%	28%	35%	40%	85%	381%
10MC-ML-SY	17%	28%	34%	40%	84%	365%
DCG on common coverage						
10M	2%	5%	5%	6%	9%	4%
10MC	2%	9%	8%	8%	9%	7%
10M-ML	5%	8%	8%	8%	9%	10%
10MC-ML	5%	7%	8%	9%	10%	9%
10M-ML-SY	7%	9%	8%	9%	10%	10%
10MC-ML-SY	8%	8%	7%	10%	12%	11%
Precision on common coverage						
10M	4%	10%	9%	10%	14%	11%
10MC	8%	13%	11%	12%	15%	11%
10M-ML	7%	10%	10%	11%	12%	12%
10MC-ML	6%	8%	10%	13%	13%	12%
10M-ML-SY	8%	11%	10%	11%	13%	14%
10MC-ML-SY	8%	11%	10%	13%	16%	16%

Questions?

The fewer the better