Learning to Rank for Quantity Consensus Queries

Somnath Banerjee   Soumen Chakrabarti
Ganesh Ramakrishnan

HP Labs India and IIT Bombay
www.cse.iitb.ac.in/~soumen/doc/QCQ

Quantity queries

- Physical quantity with units or unitless count
- Price, weight, battery life, driving time, mileage
- Frequent, commercially important query class
- For effective quantity search, must support
  - Expressing the target quantity type
  - Extracting typed quantities from text snippets
  - Assembling evidence in favor of numeric answers

```
microsoft earnings
driving time between paris and nice
battery life of lenovo x300
number of people infected by hiv worldwide
top speed of mclaren f1 car
price canon powershot sx10is
```

Sources of uncertainty

Sampling/measurement

- Height of giraffe
- Driving time from A to B
- Speed of light, value of π

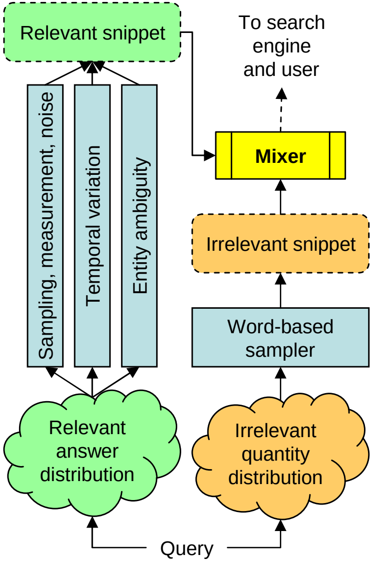
Temporal:

- Number of planets
- Pluto to Sun distance
- Microsoft revenue

Ambiguity:

- 1 ton $=$? kg
- Plutonium half-life

Snippet with incorrect quantity



Detecting consensus is nontrivial

`+giraffe, +height; foot`	
(green)	La <u>Giraffe</u> was small (approx. **11 feet** tall) because she was still young, a full grown <u>giraffe</u> can reach a <u>height</u> of **18 feet**.
(red)	<u>Giraffe</u> Photography uses a telescopic mast to elevate an 8 megapixel digital camera to a <u>height</u> of approximately **50 feet**.
(red)	The record <u>height</u> for a <u>Giraffe</u> unicycle is about **100 ft** (30.5m).
`+weight, weigh, airbus, +A380; pound`	
(green)	Since the <u>Airbus</u> <u>A380</u> <u>weighs</u> approximately **1,300,000 pounds** when fully loaded with passengers ...
(red)	The new mega-liner <u>A380</u> needs the enormous thrust of four times **70.000 pounds** in order to take off.
(red)	According to Teal, the **319-ton** <u>A380</u> would <u>weigh</u> in at **1,153 pounds** per passenger
`far +raccoon relocate; mile`	
(green)	It also says – unnervingly – that <u>relocated</u> <u>raccoons</u> have been known to return from as <u>far</u> away as **75 miles**.
(red)	Sixteen deer, 2 foxes, one skunk, and 2 <u>raccoons</u> are sighted during one **35 mile** drive.
(green)	One study found that <u>raccoons</u> could move over **20 miles** from the drop-off point in a short period of time.

- Confounding candidates with correct units
  - (four times) 70,000 pounds
  - 35 mile (drive)
  - telescopic mast . . . 50 feet
- <u>Query token</u> proximity = noisy relevance indicator
- Unit variation: 1.3 million pounds, 319 tons; 100 feet, 30.5 m

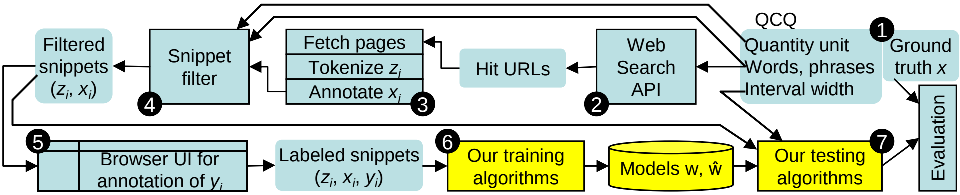
Snippet feature vector and scoring

- Snippet = window of tokens centered around quantity of desired unit/type
- Query + snippet $\longrightarrow$ feature vector z_i
- Standard TFIDF features over different fields

\+ Proximity features as used in entity ranking
  - Proximity between query token and candidate quantity = reciprocal of number of tokens between them
  - Max proximity to any query token
  - Proximity to rarest (max IDF) query token
  - IDF-weighted average proximity to all query tokens

- Relevance judgment $y_i \in \{-1, +1\}$
- From training snippets $\{(z_i, y_i)\}$ learn w
- Sort by decreasing snippet score $s_i = w^\top z_i$

QCQ system architecture



1: Query = Unit, words/phrases, interval width

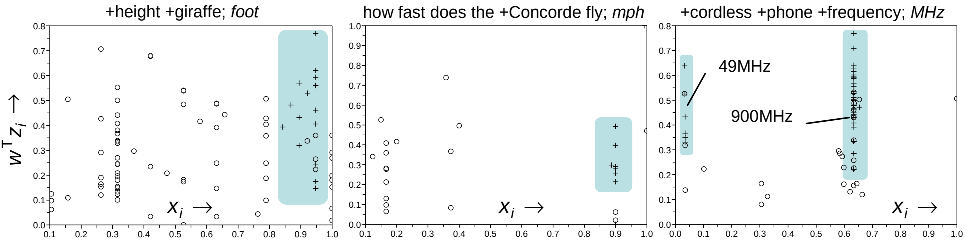
2–4: Snippet construction

- Get URL using search API, fetch pages
- Annotate quantity tokens, extract snippets
- Filter to ensure candidate quantity and ≥ 1 query tokens

5,6: Training snippet ranking model w

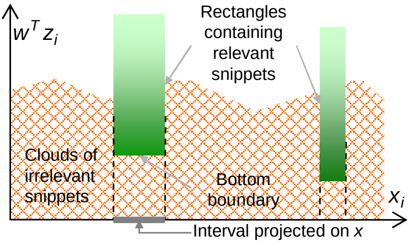
- Manually label snippets (ir)relevant
- Run Joachim's RankSVM

$w^\top z_i$ vs. x_i scatter plots



- Both axes scaled to $[0, 1]$ for clarity
- Relevant/good snippets $= +$, irrelevant/bad $= \circ$
- Ideal $w \implies$ horizontal line separating $+$ from $\circ$
- No such w for any query in our experiments
- Rectangles densely packed with many $+$, few $\circ$
  - Possibly > 1 rectangles for some queries

Consensus rectangles and intervals



- Relevant rectangle/s in sea of irrelevant snippets
- If there is any signal in $w^\top z_i$, relevant rectangles should have decent typical/average score
- But there are many low-scoring relevant snippets
- How to detect and rank consensus rectangles?
- Position and shape varies across queries
- Turns out top, bottom boundaries can be ignored

Laplacian consensus (Qin *et al.*)

- Graph with node $i \Leftrightarrow$ snippet i
- Edge $(i,j) \Leftrightarrow$ similarity between quantities x_i, x_j
- Edge weight $R(i,j)$ inversely related to $|x_i - x_j|$
  - Decay: $R(i,j) = \exp\bigl(-s(x_i - x_j)^2\bigr)$
  - Distance: $R(i,j) = \max\left\{0, 1 - \frac{|x_i - x_j|}{|x_i| + |x_j|}\right\}$
- Final score of node i is f_i

$$\text{Distortion} = \textstyle\sum_i (f_i - w^\top z_i)^2$$
$$\text{Roughness} = \textstyle\sum_{(i,j)\in E} R(i,j)(f_i - f_j)^2$$
$$\text{Violation} = \textstyle\sum_{g,b} \max\{0, 1 + f_b - f_g\} \geq \displaystyle\sum_{g,b} [\![f_g \leq f_b]\!]$$

where g, b are good, bad snippet indexes

- $\arg\min_f$ Distortion + Roughness + Violation

Wu and Marian (W&M)

- Accumulator A_x for each distinct quantity x
- Snippet (z_i, x_i) contributes score to A_{x_i}
- Snippet score decreases ...
  - Geometrically with search engine rank of containing page
  - Reciprocally with number of candidate quantities on page
  - Exponentially with number of near-duplicate pages
  - Reciprocally with distance between x_i and query tokens

⊖ Whole-page search engine rank signal inappropriate

⊖ No reinforcement between nearby quantities

⊖ Ad-hoc snippet scoring

Preliminary bake-off

		MAP	NDCG@1	NDCG@5	NDCG@10
	Web1	0.375	0.338	0.362	0.380
	Web2	0.350	0.413	0.357	0.377
	RankSVM	0.369	**0.450**	0.412	0.406
	W&M	0.306	0.247	0.303	0.322
Laplacian	Equality	0.384	0.369	0.353	0.382
Laplacian	Distance	0.407	0.413	0.401	0.420
Laplacian	Decay	**0.421**	0.433	**0.422**	**0.435**
Laplacian	Cosine	0.375	0.438	0.396	0.405

- Web1, Web2: Public search engines
  - Rewarded for correct quantity anywhere on whole page
  - Very generous upper bound to accuracy
- RankSVM: Fit w from manual per-snippet relevance judgment
- Various choices of $R(i,j)$ in Laplacian
- $\{\text{RankSVM}, \text{Laplacian-Decay}\} \succ$ W&M, others

Scanning and scoring intervals

inputs: snippets S, interval width tolerance parameter r
sort snippets S in increasing x_i order
for $i = 1, \ldots, n$ **do**
  for $j = i, \ldots, n$ **do**
    if $x_j < \left(1 + \frac{r}{100}\right) x_i$ **then**
      let $I = [x_i, x_j]$
      $merit \leftarrow GetIntervalMerit(S, I)$
      maintain intervals with top-k merit values
for surviving intervals I in decreasing merit order **do**
  present snippets in I in decreasing $w^\top z_i$ order

- Key question: how to define *GetIntervalMerit*
- $r > 0$ helps, but system robust to r

Interval merit score

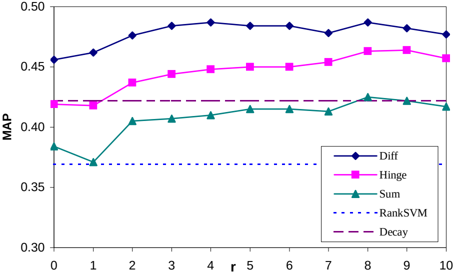
Snippet set S, quantity interval I

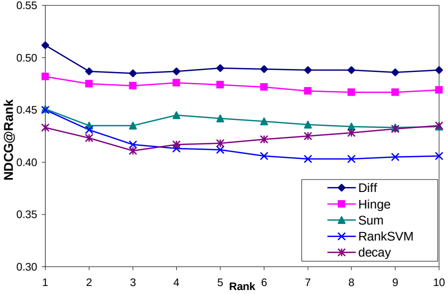
Sum: $\sum_{i:x_i \in I} w^\top z_i$ — Could have scaling problems across queries

Hinge gain: $\sum_{i:x_i \in I} \sum_{j:x_j \notin I} \max\left\{0, w^\top z_i - w^\top z_j\right\}$ — Inspired by $\textsc{RankSVM}$

Diff: $\sum_{i:x_i \in I} \sum_{j:x_j \notin I} (w^\top z_i - w^\top z_j)$ — Averaged pairwise moment

Interval merit beats all baselines

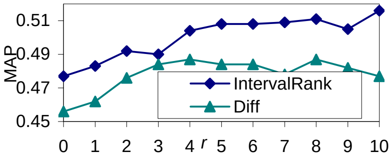


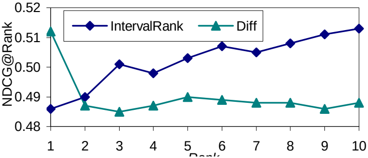


- Best for both MAP and NDCG
- Picking $r > 0$ improves accuracy
- Diff $\succ$ Hinge $\succ$ Laplacian-Decay
- Laplacian pays for $R(i,j)$ even if i,j bad
- Single width param s in $\exp(-s(x_i - x_j)^2)$ problematic

Ranking intervals directly

- I contains snippet i if $x_i \in I$
- $I \succ I'$ if I contains a larger fraction of good snippets than I'
- Invent interval features
  - All snippets in I contain {some, rarest} query word?
  - Number of distinct quantities mentioned in snippets contained in I
- Train RankSVM to order intervals
- Result: Further improvements in MAP and NDCG





Interval-oriented evaluation

- For snippet-oriented MAP and NDCG evaluation
  - First sorted intervals by decreasing merit score
  - Then reported snippets in interval by decreasing $w^\top z_i$
- Suppose we output interval list $I_1, \ldots, I_m$
  - Say I_j contains n_j snippet (quantities), k_j relevant

$$\text{IntervalPrecision@}j = (k_1 + \cdots + k_j)/(n_1 + \cdots + n_j)$$
$$\text{IntervalRecall@}j = (k_1 + \cdots + k_j)/\text{numGoodSnippets}$$

- Intuitive R-P-F1 tradeoff with interval width

	#Intervals $j \rightarrow$				
Algo, measure	1	2	3	4	5
IntervalRank recall	0.521	0.581	0.637	0.647	0.685
Laplacian-Decay recall	0.510	0.569	0.614	0.634	0.655
RankSVM recall	0.458	0.514	0.554	0.596	0.618
IntervalRank prec	0.443	0.432	0.416	0.388	0.371
Laplacian-Decay prec	0.382	0.367	0.350	0.330	0.316
RankSVM prec	0.330	0.312	0.298	0.294	0.284

Summary

- Introduced and formalized QCQs
- Standard snippet and entity ranking inadequate
- Clue from score-vs.-quantity scatter plots
- Cannot score snippet independent of others
- New collective ranking algorithms for QCQs
- Better snippet- and interval-oriented accuracy

`www.cse.iitb.ac.in/~soumen/doc/QCQ`

- $\sim$ 160 queries, $\sim$ 15000 labeled snippets available
- 500M page Web-scale evaluation in progress
- Soon: New search API with quantity support