Unsupervised Transactional Query Classification Based on Webpage Form Understanding

Yuchen Liu
Computer Science Department
University of California, Los Angeles

Xiaochuan Ni, Jian-Tao Sun, Zheng Chen
Machine Learning Group
Microsoft Research Asia
Content

• Motivation and Problem Definition
• Data Explanation
• Solution
• More Discussion
• Experiments
• Conclusion and Future Works
Content

• Motivation and Problem Definition
• Data Explanation
• Solution
• More Discussion
• Experiments
• Conclusion and Future Works
Motivation

- Search Query Type Classification [Broder-2002]:
  - Informational Queries: (~39%) “canon 5d review”
  - Navigational Queries: (~25%) “myspace”
  - Transactional Queries: (~36%) “book Marriott hotel for my Seattle visit”
Problem Definition

• Transactional Query Classification: binary classification problem

• Common transactions:
  - book flights
  - purchase movie tickets
  - find car dealers

• Query Examples:
  - “book flight from Los Angeles to Glasgow”
  - “buy movie tickets for Lion King 3D”
  - “find dealers for Honda around Seattle”
Previous Works

• Query Type Classification
  • Learning-based: focus on informational and navigational queries
    - Transactional queries not addressed well
    - Require cumbersome human labeling efforts
  • Rule-based: require human-compiled rules, not scalable
Observation

• What do users do after typing a transactional query into search box?
Want to simplify users’ transaction by “auto-completion and delivery of forms”? 

Sales and Airfare Deals - AirTran.com
Flint: $98 - $199; Ft. Lauderdale: $71 - $199; Ft. Myers ... imposed September 11th Security Fee of up to $5 one-way. ... San Juan, Puerto Rico and International travel excluded.
www.airtran.com/sale
Our Focus

- Unsupervised Approach for Transactional Query Classification
  - Making use of form clicking signal for queries
  - Build classifier with the help of information in webpage forms
  - Reduce the human labeling effort for training classifier
Challenges: Identifying Transactional Queries

- Not all queries which are often associated with form clicking are transactional queries.
  - Some navigational queries: “american airlines”
  
- Ranking queries to estimate how transactional they are.

- Not all forms are served as an entrance to a transaction.
  - “Search box” issue

- Mutual reinforcement principle
Challenges: Building the Classifier

- How to deal with queries which we haven’t seen in the log?
  - Generate query patterns for further classification.
Pattern Generalization Example

cheap airfare from san juan to flint one way

cheap airfare from [from] to [to] [travel type]
[from]: Atlanta GA, Boston MA, ...
[to]: Allentown PA, Cancun Mexico, ...
[travel type]: round trip, one-way
Content

• Motivation and Problem Definition

• Data Explanation

• Solution

• More Discussion

• Experiments

• Conclusion and Future Works
Webpage Form Modeling

- Everything between html tag “<form>” and </form>”.

**Slot:** (label, type, values)
  - **label:** description of the slot
  - **type:** text, select, radio, checkbox, button
  - **values:** pre-set value list

**Form:** (label, slots, from-url, to-url)
  - **label:** description of the form
  - **slots:** collection of all slots
  - **from-url:** the url of webpage containing this form
  - **to-url:** the url which users will visit after submission
Webpage Form Example

Form1
- **From-url**: http://www.united.com/
- **To-url**: http://www.ua2go.com/flifo/FlightSummary.do
- **Label**: check status of a flight

**Slot1**
- **Type**: Select
- **Label**: date
- **Values**: (07/25/2010, ...)

**Slot2**
- **Type**: Text
- **Label**: flight number
- **Values**: none

**Slot3** (Text. from. none)

**Slot4** (Text. to, none)

**Slot5**
- **Type**: Select
- **Label**: time
- **Values**: (all day, ...)

**Slot6**
- **Type**: Button
- **Label**: check
- **Values**: none
Associate Query with Form

- Toolbar Log
  - Collected by IE toolbar, recording users’ behavior.
  - Format: `{userid, search query, clicked result page URL, further clicked page URL in result page}`

- Building **query-form** graph: bipartite graph
  - matching of `{from-url, to-url}` in form model with `{clicked result page URL, further clicked page URL}` in toolbar log
Content

- Motivation and Problem Definition
- Data Explanation
- Solution
- More Discussion
- Experiments
- Conclusion and Future Works
Framework

• Three Components:

1. Analyze query-form graph to get high-quality transactional queries as **training data**.

2. **Generalize query patterns** using training data with corresponding web forms.

3. **Predict query types** adopting query patterns for new queries.
STEP I: Collecting Training Data

- Idea: ranking queries by how transactional they are.
  - two scores: click entropy & click ratio
STEP 1: Collecting Training Data (cont’d)

- Click Entropy

- Consider click distribution over both websites and forms.

  “flights from ny to la”

  “american airlines”

  “book flights”

  typical transactional  navigational  good transactional

- We prefer giving higher score to $q_c$, then $q_a$, than $q_b$. 
STEP 1: Collecting Training Data (cont’d)

- Click Entropy

\[
ClickEntropy(q) = (1 + E(p_{site})) \sum_{s_i \in Site(q)} \frac{Click(q, s_i)}{Click(q)} \frac{1}{2^{E(p_{s_i})}}
\]  

- \(E(p)\) is entropy function for distribution \(p\).
STEP 1: Collecting Training Data (cont’d)

• Click Ratio

• how often do users click on forms after submitting this query?

\[ \text{ClickRatio}(q) = \log_2 \left( \frac{\text{Click}(q)}{\text{Impression}(q)} + 1 \right) \]  

(3)

• Impression(q) is total appearance of q in log.
STEP I: Collecting Training Data (cont’d)

• Ranking queries

\[ \text{Score}(q) = \text{ClickEntropy}(q_i) \times \text{ClickRatio}(q_i) \]  

• to estimate how likely a query is transactional.
STEP2: Query Pattern Generalization

- **Query pattern**: a sequence of terms (including at least one slot)
  - example: “used car [make] [model]”
- Use query-form pair to generalize the query into a query pattern
  - Input: query (q) & a list of slots from form (f)
  - Output: a query pattern if applicable
STEP2: Query Pattern Generalization (cont’d)

- Idea: if we find a value of a slot appears in the query, replace the value with the slot in the query to generalize a query pattern.

- Mis-spelling - fuzzy string match for value matching.

- Greedy algorithm for replacing values with slots.

- Use redundancy to rank query patterns by score:

\[
Score_{QP}(p) = |Slot(p)| \sum_{q_i \in Q(p)} Score(q_i)
\]  

(6)
Pattern Generalization Example

cheap airfare from [from] to [to] [travel type]
[from]: Atlanta GA, Boston MA, ...
[to]: Allentown PA, Cancun Mexico, ...
[travel type]: round trip, one-way
STEP3: Query Classification

- Idea: Use query patterns to classify a new query
  
  - Calculate how a query \( (p) \) fits a query pattern \( (p) \)
    
    1. conduct pattern generalization algorithm to \( q \) with slots in pattern \( p \)
    
    2. calculate fit score by what they have left
      \[
      \text{Fit}(p, q) = \frac{\text{SlotMatch}(p, q)}{|\text{Slot}(p)|} \cdot \text{Cosine}(p', q')
      \]
    
    - Find the query pattern which fits the query best
    
    - Compute the transactional score for the query \( (q) \)
      \[
      \text{Score}_Q(q) = \text{Fit}(p_{fit}, q) \times \text{Score}_{QP}(p_{fit}) \quad (8)
      \]
Content

• Motivation and Problem Definition

• Data Explanation

• Solution

• More Discussion

• Experiments

• Conclusion and Future Works
Improving Training Data Quality

• “mutual reinforcement principle”:
  - a transactional query is usually associated with transactional forms, a transactional form is usually associated with transactional queries.

• iterative algorithm for updating transactional score for both queries and forms

\[
Score_{k+1}(q_i) = \text{ClickEntropy}(q_i) \times \text{ClickRatio}(q_i) \times \sum_{f_j \in F(q_i)} \text{Click}(q_i, f_j)Score_k(f_j) \tag{10}
\]

\[
Score_{k+1}(f_j) = \sum_{q_i \in Q(f_j)} \text{Click}(q_i, f_j)Score_k(q_i) \tag{11}
\]
Improving Transactional Query Pattern Quality

- **quality of slots** (completeness of values, slots) affects pattern quality

- grouping forms with same functionality in different websites together could give us **more slots**, and **more values** in each slot.

- two-step clustering algorithm for forms
Improving Transactional Query Pattern Quality (cont’d)

- Two-step clustering algorithm for forms:

  1. clustering slots with same functionality together.

     \[
     \text{Sim}(s_i, s_j) = \alpha L \text{Sim}(s_i, s_j) + \beta V \text{Sim}(s_i, s_j) \\
     +(1 - \alpha - \beta) \text{CoSim}(s_i, s_j)
     \]  
     \[ (12) \]

  2. clustering forms using slot clustering result.

     \[
     \text{Sim}(f_i, f_j) = \gamma L \text{Sim}(f_i, f_j) + (1 - \gamma) S \text{Sim}(f_i, f_j)
     \]  
     \[ (13) \]
Content

• Motivation and Problem Definition
• Data Explanation
• Solution
• More Discussion
• Experiments
• Conclusion and Future Works
Experiments

- Experimental Settings

  - 100 million log records, 30 million uni. queries, 7 million uni. users

- Crawled forms from 3 categories “flight”, “hotel”, “car”.
  - our solution could apply for any other domain.

- Query-form Graph: 155 forms with 20,934 queries.
Training Data Collection Evaluation

- Manually labeled 3000 queries pooled by both methods
- Top K Precision

![Graph 1](image1.png)

*Figure 5: Top K Precision of Transactional Query Identification*

- Convergence of Iterative Algorithm

![Graph 2](image2.png)

*Figure 6: Top 250 Precision with Iteration Number*
Query Pattern Generation Evaluation

- Manually labeled top 50 query patterns generated by each methods into 4 ratings.
- How slot clustering threshold affects query pattern quality and quantity:
- How form clustering threshold affects query pattern quality and quantity:
Query Pattern Generation Evaluation (cont’d)

• Comparison of different methods:

![Diagram showing comparison of different methods: Direct-match, Slot-cluster-based, Form-cluster-based. The diagram illustrates the average quality and number of patterns generated for each method. The x-axis represents the pattern generation method, and the y-axis shows the top 50 pattern average quality and number of patterns. The results indicate that Form-cluster-based method has the highest average quality and number of patterns.](image)

Figure 9: Comparison of Different Methods

36 | CIKM 2011
Query Pattern Generation Evaluation (cont’d)

- Generalized query pattern examples:

<table>
<thead>
<tr>
<th>Generalized Query Patterns with High Confidence Score</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Flight Domain</strong></td>
</tr>
<tr>
<td>flights from [departure] to [destination]</td>
</tr>
<tr>
<td>[departure] to [destination]</td>
</tr>
<tr>
<td>flights to [destination]</td>
</tr>
<tr>
<td>cheap flights from [departure] to [destination]</td>
</tr>
<tr>
<td>cheap flights to [destination]</td>
</tr>
<tr>
<td>flights [departure] to [destination]</td>
</tr>
<tr>
<td>[departure] airport</td>
</tr>
<tr>
<td>flights to [destination] from [departure]</td>
</tr>
<tr>
<td>airlines that fly to [destination]</td>
</tr>
<tr>
<td>[maximum stops] flights to [destination]</td>
</tr>
</tbody>
</table>

Table 3: Query Patterns Generalized using Form Clusters
Transactional Query Prediction Evaluation

• We compare our method with a supervised learning approach
  - trained a SVM binary classifier with uni-gram features
• F1: 83% - comparable F-Measure with supervised learning approach
  - but without any human labeling process

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern-based</td>
<td>0.832</td>
<td>0.830</td>
<td>0.830</td>
</tr>
<tr>
<td>Supervised-approach</td>
<td>0.864</td>
<td>0.848</td>
<td>0.847</td>
</tr>
</tbody>
</table>
Feature for Supervised Learning

• Using pattern-matching as a feature in supervised learning method
  – different percentage of training data for training process
Content

• Motivation and Problem Definition
• Data Explanation
• Solution
• More Discussion
• Experiments
• Conclusion and Future Works
Conclusion

• This unsupervised classification method could provide a comparable performance with simple supervised learning approach without any human labeling efforts.

• Pattern matching could provide help for supervised learning approach, especially when training dataset is small.

• Query patterns provide us a better understanding of how users formulate their transactional queries.
Future Works

• Optimize search engines’ ranking for transactional queries by considering users’ intent to conduct transactions on webpage forms.

• Directly provide users with the forms they might click for transactional queries instead of ten blue links.

• Auto form-filling based on a better structured understanding of users’ transactional queries.
Thanks!

Any Questions?