Automatically Identifying Localizable Queries

Michael Welch, Junghoo Cho
UCLA Computer Science Department
Localizable Queries

• Some queries are location sensitive
  – “italian restaurant” ➔ “[city] italian restaurant”
  – “courthouse” ➔ “[county] courthouse”
  – “drivers license” ➔ “[state] drivers license”

• Our task: identify this class of queries
Motivation

• Why automatically localize?
  – Reduce burden on the user
    • No special “local” or “mobile” site
  – Improve search result relevance
    • Not all information is relevant to every user
  – Increase clickthrough rate
  – Improve local sponsored content matching
Motivation

• Significant fraction of queries are localizable
  – Roughly 30%, but users only explicitly localize them about $\frac{1}{2}$ of the time

• Users exhibit consensus on which queries are localizable
Our Approach

• Identify candidate localizable queries
• Select a set of relevant features
• Train and evaluate supervised classifier performance
Keep It Simple

• General principle: keep it simple
  – We’re dealing with web scale data
• Independent processing stages
• Features should be easy to compute
  – Distributable, in parallel
Our Approach

• Identify candidate localizable queries
• Select a set of relevant features
• Train and evaluate supervised classifier performance
Identifying Base Queries

• Queries are short and unformatted
• Use string matching
  – Compare against locations of interest
    • Using U.S. Census Bureau data
  – Tag matching parts and extract the “base”
  – Filter out false positives in the classifier
  – Simple, yet effective
Example: Identifying Base Queries

- city: malibu
  - public libraries in california
  - state: california
    - public libraries in california
      - state: california
        - public libraries in
      - public libraries in malibu
        - city: malibu
          - public libraries in
Example: Identifying Base Queries

- Three distinct base queries
  - Remove stop words and group by base
  - Allows us to compute aggregate statistics

<table>
<thead>
<tr>
<th>Base</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>public libraries california</td>
<td>city:malibu</td>
</tr>
<tr>
<td>public libraries malibu</td>
<td>state:california</td>
</tr>
<tr>
<td>public libraries</td>
<td>city:malibu, state:california</td>
</tr>
</tbody>
</table>
Our Approach

• Identify candidate localizable queries
• **Select a set of relevant features**
• Train and evaluate supervised classifier performance
Distinguishing Features

• Hypothesis: localizable queries should
  – Be explicitly localized by some users
  – Occur several times
    • From different users
  – Occur with several different locations
    • Each with about equal probability
Localization Ratio

- Users vote for the localizability of query $q_i$ by contextualizing it with a location $L$

$$r_i = \frac{Q_i(L)}{Q_i + Q_i(L)}$$

- Susceptible to small sample sizes
Occurrence Counts

• Measure overall popularity of query
  – Not necessarily indicative of localizability
  – Can be used to normalize other measures

• User-related counts
  – Users often issue same query multiple times
  – Unique user count is a better measure of popularity for our purpose

• Location counts
  – Number of distinct locations
Location Distribution

- The “fried chicken” problem

<table>
<thead>
<tr>
<th>Tag</th>
<th>Count</th>
<th>Tag</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>city:chester</td>
<td>6</td>
<td>city:rice</td>
<td>2</td>
</tr>
<tr>
<td>city:colorado springs</td>
<td>1</td>
<td>city:waxahachie</td>
<td>1</td>
</tr>
<tr>
<td>city:cook</td>
<td>1</td>
<td>state:kentucky</td>
<td>163</td>
</tr>
<tr>
<td>city:crown</td>
<td>1</td>
<td>state:louisiana</td>
<td>4</td>
</tr>
<tr>
<td>city:louisiana</td>
<td>4</td>
<td>state:maryland</td>
<td>2</td>
</tr>
<tr>
<td>city:louisville</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Location Distribution

- The “fried chicken” problem

<table>
<thead>
<tr>
<th>Tag</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>city:chester</td>
<td>2</td>
</tr>
<tr>
<td>city:colorado</td>
<td>1</td>
</tr>
<tr>
<td>city:cook</td>
<td>163</td>
</tr>
<tr>
<td>city:crown</td>
<td>4</td>
</tr>
<tr>
<td>city:en</td>
<td>2</td>
</tr>
<tr>
<td>city:family</td>
<td>2</td>
</tr>
<tr>
<td>city:hayden</td>
<td>2</td>
</tr>
</tbody>
</table>

KFC
\[ \forall \ell \in L(q_b) \Pr[\ell \in q_\ell \mid q_b = base(q_\ell)] \approx \frac{1}{|L(q_b)|} \]

- Informally: given any instance of a localized query \(q_l\) with base \(q_b\), the probability that \(q_l\) contains location \(\ell\) is approximately uniform across all locations that occur with \(q_b\).
- Approximate the distribution with mean, median, min, max, and standard deviation
Clickthrough Rates

• Assumption: greater clickthrough rate indicative of higher user satisfaction

• Calculated clickthrough rates for both the base query and its localized forms
  – Binary clickthrough function

• Clickthrough rate for localized instances 17% higher than nonlocalized instances
Our Approach

- Identify candidate localizable queries
- Select a set of relevant features
- Train and evaluate supervised classifier performance
Classifier Training Data

• Selected a random sample of 200 base queries generated by the tagging step
• Filtered out base queries where
  – $n_L \leq 1$
  – $u_q = 1$
  – $q = 0$
• From remaining 102 queries
  – 48 positive (localizable) examples
  – 54 negative examples
Evaluation Setup

- Evaluated supervised classifiers on precision and recall using 10-fold cross validation
  - Precision: accuracy of queries classified as localizable
  - Recall: percent of localizable queries identified
- Focused attention on *positive* precision
  - False positives more harmful than false negatives
  - Recall scores account for manual filtering
Individual Classifiers

- Naïve Bayes
  - Gaussian assumption doesn’t hold for all features
- Decision Trees
  - Emphasised localization ratio, location distribution measures, and clickthrough rates

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve Bayes</td>
<td>64%</td>
<td>43%</td>
</tr>
<tr>
<td>Decision Tree (Information Gain)</td>
<td>67%</td>
<td>57%</td>
</tr>
<tr>
<td>Decision Tree (Normalized Information Gain)</td>
<td>64%</td>
<td>56%</td>
</tr>
<tr>
<td>Decision Tree (Gini Coefficient)</td>
<td>68%</td>
<td>51%</td>
</tr>
</tbody>
</table>
Individual Classifiers

- SVM
  - Improvement over NB and DT, but opaque
- Neural Network
  - Also opaque
  - Best individual classifier

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>75%</td>
<td>62%</td>
</tr>
<tr>
<td>Neural Network</td>
<td>85%</td>
<td>52%</td>
</tr>
</tbody>
</table>
Ensemble Classifiers

• Observation: false positive classifications differ for individual classifiers
• Combined DT, SVM, and NN using a majority voting scheme
• Achieved 94% precision with 46% recall
Main Contributions

• Method for classifying queries as localizable
  – Scalable, language independent tagging
  – Determined useful features for classification
  – Demonstrated simple components can make a highly accurate system

• Exploited variation in classifiers by applying majority voting
Future Work

• Optimize feature computation for real-time
  – Many features fit into MapReduce framework

• Investigate using dynamic features
  – Updating classifier models
  – Explicit feedback loops

• Generalize definition of “location”
  – Landmarks, relative locations, GPS

• Integration with search system
Acknowledgements

• Anonymous reviewers and survey participants provided valuable data and feedback

• Generous travel support provided by
  – ACM SIGIR
  – Amit Singhal, in honor of Donald B. Crouch
  – Microsoft Research, in honor of Karen Sparck Jones.
Questions or Comments?

... and hopefully some answers