Disjunctive Sets of Phrase Queries for Diverse Query Suggestion

Ziyang Liao, Keishi Tajima
Kyoto University

WI 2019
2019.10.16
Many ambiguous queries that can be interpreted in multiple ways

Query
orange fertilizer

How to give orange trees fertilizer  Popular Intent
How to make oranges into fertilizer  Infrequent Intent

Query results are filled with pages corresponding to popular query intents

Difficult to find queries that can retrieve pages corresponding to infrequent query intents.
Existing method 1

- Search engines recommend queries extracted from query log data
  - Queries for infrequent intents appear in the query log data only infrequently

E.g., Bing query recommendation result for “Orange fertilizer”

関連キーワード

orange trees fertilizer
fertilizer for orange trees florida
natural fertilizer for orange trees
satsuma orange fertilizer
best shrub fertilizer
types of fertilizers for crops
alternatives to nitrogen fertilizer
fertilizer plant

How to make orange into fertilizer ✗

How to give orange tree fertilizer ✔
**Existing method 2**

- Extract phrases that represent relationships between query terms from a web corpus [1]
  - Web corpus includes more information than query log data
  - It is still difficult to find phrases corresponding to infrequent query intents
  - Phrases corresponding to infrequent query intents are buried in those corresponding to popular query intents

[1] Daisuke Fukuchi, Takehiro Yamamoto, Katsumi Tanaka, Query Mining Based on Term Relationship Estimation in Verbal Queries, Transactions of the Japanese Society for Artificial Intelligence, 2017, Volume 32, Issue 1, Pages WII-J_1-15,
Our Approach 1

- Expand query terms using hypernyms, sister terms, and hyponyms

To increase the chance of finding phrases corresponding to infrequent query intents, we expand each query term into the set of its hypernyms, sister terms, and hyponyms.

- hypernym
- sister term
- hyponym
Our Approach 1 example

- Expand the query term “orange”
- Find connecting phrases using all expanded query terms

Query: Orange Fertilizer
Expansion: Orange → {Orange, Fruit, ...}
Phrase: Fruit can replace Fertilizer

Expanded query: “Orange can replace Fertilizer”
Approach 2

- Clustering extracted phrases
  - By query term expansion, we obtain many phrases
  - Phrases for infrequent intents are buried in those for popular intents
  - Cluster related phrases and only show exemplars of the clusters

Clusters:
- is
- was
- has
- had
- get
- got
- receive

Selected by user:
- “Orange is fertilizer” or “Orange was fertilizer”
Overview

User Input:
orange fertilizer

Suggestions:
- orange need fertilizer
- orange provide fertilizer
- orange can replace fertilizer

Tokenization

Word expansion

Orange

WordNet

Fertilizer

Word expansion

Expanded T1 term set

Expanded T2 term set

Clustering

R4

R6

provide

Exemplar

need

R2

R8

Exemplar

can replace

Exemplar

can replace

Web Corpus

Relation Extraction

need

provide

...
Method details

- **Query term expansion**
  - WordNet

- **Connecting phrase extraction**
  - OpenIE (Open information extraction) [2]

- **Web corpus**
  - ClueWeb

- **Clustering algorithm**
  - Affinity Propagation algorithm

Open Information Extraction

- Word extraction tool
  - OpenIE (Open information extraction)

Born in a small town, she took the midnight train going anywhere.

(input)

(she; took; midnight train)
Data processing procedure

- **Remove triples with incorrect information**
  - Triple with typographical errors (eg: apple ai fertilizer)
  - 0 results in search engine (eg: apple is computed from fertilizer)

- **Similarity calculation and clustering of related phrases**
  - Extracting related phrases from the captured relationship triples
  - Calculate similarity between related phrases
  - Clustering based on similarity

- **Calculate the overall weight of the cluster and rank**
  - Score considering the frequency of elements in the cluster and the type of expansion
Clustering algorithm

- **Affinity Propagation algorithm**
  - Premise
    - Cluster target: phrases
    - Distance: Value of the semantic similarity between phrases
  - Nature
    - Dynamically determines the number of clusters without the need to specify K as in K-means
    - A point to be exemplar is chosen from each cluster
Cluster ranking algorithm

- **Ranking score**
  
  Query: “e1 e2”

  \[
  S(c_i) = w(e_i^1)w(e_i^2) \sum_{p_j \in c_i} f_j
  \]

  - \( S(c_i) \): cluster \( c_i \) ranking score
  - \( w(e_i^1) \): the weight of the expansion type of e1
  - \( w(e_i^2) \): the weight of the expansion type of e2
  - \( f_j \): The frequency of phrase \( p_j \) in cluster \( c_i \)

<table>
<thead>
<tr>
<th>Expansion type e</th>
<th>Weight value w(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>hypernym</td>
<td>1</td>
</tr>
<tr>
<td>sister terms</td>
<td>2</td>
</tr>
<tr>
<td>itself &amp; hyponym</td>
<td>3</td>
</tr>
</tbody>
</table>
For “orange is fertilizer”

- The connecting phrase “is” is extracted by using “orange” itself and a hypernym of “fertilizer”
- The cluster size of “is” is 10

- \( S(c_i) \): cluster \( c_i \) ‘s ranking score
- \( w(e^1_i) \): The weight of the expansion type of \( e1 = 3 \)
- \( w(e^2_i) \): The weight of the expansion type of \( e2 = 1 \)
- \( f_j \): The frequency of phrase \( p_j \) in cluster \( c_i = 10 \)

\[
S(c_i) = w(e^1_i)w(e^2_i) \sum_{p_j \in c_i} f_j = 3 \times 1 \times 10 = 30
\]
Experiment

Procedure

1. Prepares queries that have multiple interpretations

2. Get query recommendation results
   - Query recommendation results of Google
   - Query recommendation results of Bing
   - Query recommendation results of proposed method

3. Search the web by using query recommendation results with Bing and get the top 10 results

4. Calculate the precision of the results that match the infrequent intent
### Test queries

Prepares multiple interpretable queries

<table>
<thead>
<tr>
<th>Id</th>
<th>Query</th>
<th>Common search intent</th>
<th>Infrequent search intent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>apple fertilizer</td>
<td>Fertilizer for apple</td>
<td>How to change apple into fertilizer</td>
</tr>
<tr>
<td>1-2</td>
<td>banana fertilizer</td>
<td>Fertilizer for banana</td>
<td>How to change banana into fertilizer</td>
</tr>
<tr>
<td>1-3</td>
<td>orange fertilizer</td>
<td>Fertilizer for orange</td>
<td>How to change orange into fertilizer</td>
</tr>
<tr>
<td>2-1</td>
<td>Kyoto bank</td>
<td>The Kyoto Bank</td>
<td>Information about the banks in Kyoto</td>
</tr>
<tr>
<td>2-2</td>
<td>Japan bank</td>
<td>The Bank of Japan</td>
<td>Information about the banks in Japan</td>
</tr>
<tr>
<td>2-3</td>
<td>China bank</td>
<td>The China Bank</td>
<td>Information about the banks in China</td>
</tr>
<tr>
<td>3-1</td>
<td>Steak sauce</td>
<td>Sauce for steak</td>
<td>How to make sauce using steak</td>
</tr>
<tr>
<td>3-2</td>
<td>Beef sauce</td>
<td>Sauce for beef</td>
<td>How to make sauce using beef</td>
</tr>
<tr>
<td>3-3</td>
<td>Chicken sauce</td>
<td>Sauce for chicken</td>
<td>How to make sauce using chicken</td>
</tr>
</tbody>
</table>
### Get query recommendation results

Result of Google query recommendation and proposed method

<table>
<thead>
<tr>
<th>Google</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>orange tree fertilizer florida</td>
<td>orange is fertilizer</td>
</tr>
<tr>
<td>homemade fertilizer for citrus trees</td>
<td>fertilizer produced orange</td>
</tr>
<tr>
<td>citrus fertilizer</td>
<td>fertilizer transported in orange</td>
</tr>
<tr>
<td>how often to water orange trees</td>
<td>fertilizer end up in orange</td>
</tr>
<tr>
<td>when to fertilize citrus trees in southern california</td>
<td>orange contains fertilizer</td>
</tr>
<tr>
<td>citrus fertilizer npk</td>
<td>fertilizer arrow orange</td>
</tr>
<tr>
<td>liquid citrus fertilizer with micronutrients</td>
<td>fertilizer pollute orange</td>
</tr>
<tr>
<td>fertilizer production</td>
<td>fertilizer found in orange</td>
</tr>
<tr>
<td></td>
<td>orange has fertilizer</td>
</tr>
<tr>
<td></td>
<td>orange made from fertilizer</td>
</tr>
</tbody>
</table>

*Not match for infrequent intents*  
*Match for infrequent intents*
## Patterns

We calculate the result of following patterns

<table>
<thead>
<tr>
<th>Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Synonym expansion</td>
</tr>
<tr>
<td>2) Typeof expansion</td>
</tr>
<tr>
<td>3) Instanceof expansion</td>
</tr>
<tr>
<td>4) Hastype expansion</td>
</tr>
<tr>
<td>5) Hasinstance expansion</td>
</tr>
<tr>
<td>6) Full expansion without expansion type scoring</td>
</tr>
<tr>
<td>7) Full expansion with expansion type scoring</td>
</tr>
</tbody>
</table>
## Result

Calculate precision of results that match minor intentions

<table>
<thead>
<tr>
<th>id</th>
<th>1-1</th>
<th>1-2</th>
<th>1-3</th>
<th>2-1</th>
<th>2-2</th>
<th>2-3</th>
<th>3-1</th>
<th>3-2</th>
<th>3-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>0.00</td>
<td>0.14</td>
<td>0.00</td>
<td>0.00</td>
<td>0.08</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Bing</td>
<td>0.00</td>
<td>0.24</td>
<td>0.00</td>
<td>0.03</td>
<td>0.04</td>
<td>0.00</td>
<td>0.00</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>Original query</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.07</td>
<td>0.05</td>
<td>0.02</td>
<td>0.00</td>
<td>0.03</td>
</tr>
<tr>
<td>1) Synonym expansion</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2) Typeof expansion</td>
<td>0.00</td>
<td>0.38</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3) Instanceof expansion</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4) Hastype expansion</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5) Hasinstance expansion</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>6) Full expansion without expansion type scoring</td>
<td>0.05</td>
<td>0.41</td>
<td>0.06</td>
<td>0.00</td>
<td>0.09</td>
<td>0.07</td>
<td>0.02</td>
<td>0.05</td>
<td>0.03</td>
</tr>
<tr>
<td>7) Full expansion with expansion type scoring</td>
<td>0.05</td>
<td>0.41</td>
<td>0.06</td>
<td>0.00</td>
<td>0.10</td>
<td>0.09</td>
<td>0.03</td>
<td>0.10</td>
<td>0.03</td>
</tr>
</tbody>
</table>
Propose query expansion method for infrequent query intents

- To obtain more candidate queries, expand the query terms, and extract phrases that connect the two query terms.
- A large number of phrases are extracted from the corpus and clustered to generate diverse phrases.
- Add connecting phrases to the original query and generate query candidates.
- The experimental results show the usefulness of the proposed method.