Ranking Methods for Query Relaxation in Book Search

Momo Kyozuka and Keishi Tajima
Kyoto University, Japan
Background

- Book search situation
- Users only have vague memories of the stories
- Queries may include wrong keywords

a boy makes paints with blue seeds

Search

flowers
Approach

Problem
• Make new queries by removing wrong keywords

→ Cannot know which words are wrong

Solution
• Use every subset as a new query
• Rank according to reliability of the subsets
Proposed method

A sentence query from a user
In the book,
a boy makes paints with blue seeds.

Rank subset queries based on reliability of query words

<table>
<thead>
<tr>
<th></th>
<th>a boy</th>
<th>make</th>
<th>paint</th>
<th>blue</th>
<th>seed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>a boy</td>
<td></td>
<td>paint</td>
<td></td>
<td>seed</td>
</tr>
<tr>
<td>3</td>
<td>a boy</td>
<td></td>
<td>paint</td>
<td>blue</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>seed</td>
</tr>
</tbody>
</table>

Generate all subset queries

<table>
<thead>
<tr>
<th>a boy</th>
<th>make</th>
<th>paint</th>
<th>blue</th>
<th>seed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>make</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a boy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Concatenate results

Result of query 1
1. Book 1
2. Book 2
3. Book 3

Result of query 2
1. Book 4
2. Book 5
3. Book 6

Result of query 3
1. Book 7
2. Book 8

...
Hypothesis

Reliability of each word depends on its semantic role

- Collect sentence queries from users
- Parse dependencies of the sentence query and classify words into four roles

A sentence query from a user

In the book,

a boy makes paints with blue seeds.

Classify query words

- a boy
  - subject
- makes
  - predicate
- paints
  - object
- blue
  - others
- seeds
  - object
Data used in our experiments

• 37 sentence queries by users
  • collected from Yahoo! Chiebukuro (a Japanese popular QA site)
  • question and answer data
    • Q : users asked for titles of books based on vague memories
    • A : questioners found that the answer books were correct

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
<th>Comment from questioner</th>
</tr>
</thead>
<tbody>
<tr>
<td>I am searching for a book that I read when I was a child. In the book, a boy makes paints with blue seeds.</td>
<td>It may be wrong, but I think the book is “Real Sky-Blue”. You can see the design of the book in this web page.</td>
<td>I checked the web page and I found that the book is what I was searching for. Thank you very much.</td>
</tr>
</tbody>
</table>

• Correct descriptions of target books
  • collected from the database of National Diet Library of Japan (NDL)
Which semantic role is more reliable?

- Examine whether each word in the sentence queries also appears in the database description.
- Calculate the ratio of the appearing words in both descriptions for each of the four roles.
- Use each ratio as the probability that a query word in each class also appears in the correct description of the target book.

<table>
<thead>
<tr>
<th>class</th>
<th>in user description</th>
<th>in database description</th>
<th>ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>43</td>
<td>19</td>
<td>0.442</td>
</tr>
<tr>
<td>predicate</td>
<td>62</td>
<td>3</td>
<td>0.048</td>
</tr>
<tr>
<td>object</td>
<td>55</td>
<td>30</td>
<td>0.545</td>
</tr>
<tr>
<td>others</td>
<td>34</td>
<td>15</td>
<td>0.441</td>
</tr>
</tbody>
</table>

The reliability of predicate is lowest.
Ranking queries

• Proposed method: Probability Ranking Principle (PRP)
  • Classify query words according to their reliability
  • Rank queries by their reliability
• Baseline 1: Number of Words method
  • Rank queries by number of included words
• Baseline 2: TF-IDF
  • Rank queries by the cosine similarity to the original query in the descending order
Probability Ranking Principle (PRP)

- $QP(q)$: the probability that all words in a generated query $q$ appear in the description in the database
  - Defined as the product of the reliability value of each word in the query

**e.g.**

$q =\begin{array}{c}
\text{a boy} \\
\text{subject}
\end{array} \begin{array}{c}
\text{paints} \\
\text{object}
\end{array} \begin{array}{c}
\text{seeds} \\
\text{object}
\end{array}$

$QP(q) = 0.442 \times 0.545 \times 0.545 = 0.131$

- The probability that a word in subject class appears in the correct description is 0.442
- The probability that a word in object class appears in the correct description is 0.545
Probability Ranking Principle (PRP)

- $hit(q)$: the number of search results for the query $q$
  - ex. $hit(q) = 5$

- $p(q)$: the probability that each book in the search results is correct
  - Define $p(q)$ as $QP(q)/hit(q)$
  - ex. $p(q) = QP(q)/hit(q) = 0.131/5 = 0.026$

- Ranked queries by $p(q)$ in descending order

![Search Result]

- a boy paints seeds
- Number of Search Result: 5
- $QP(q) = 0.131$
Baseline 1: Number of Words method

- Rank the queries by the number of included words in the descending order

A sentence query from a user
In the book,

**a boy makes paints with blue seeds.**

<table>
<thead>
<tr>
<th></th>
<th>a boy</th>
<th>makes</th>
<th>paints</th>
<th>blue</th>
<th>seeds</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a boy</td>
<td></td>
<td>paints</td>
<td>blue</td>
<td>seeds</td>
</tr>
<tr>
<td>2</td>
<td>a boy</td>
<td>makes</td>
<td></td>
<td>blue</td>
<td>seeds</td>
</tr>
<tr>
<td>3</td>
<td>makes</td>
<td></td>
<td>paints</td>
<td>blue</td>
<td>seeds</td>
</tr>
<tr>
<td>4</td>
<td>a boy</td>
<td>makes</td>
<td>paints</td>
<td></td>
<td>seeds</td>
</tr>
<tr>
<td>5</td>
<td>a boy</td>
<td>makes</td>
<td></td>
<td>blue</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>a boy</td>
<td></td>
<td></td>
<td>blue</td>
<td>seeds</td>
</tr>
<tr>
<td>7</td>
<td>makes</td>
<td></td>
<td></td>
<td>blue</td>
<td>seeds</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>14</td>
<td>a boy</td>
<td>makes</td>
<td>paints</td>
<td></td>
<td>seeds</td>
</tr>
<tr>
<td>15</td>
<td>a boy</td>
<td>makes</td>
<td>paints</td>
<td></td>
<td>blue</td>
</tr>
</tbody>
</table>

- Queries composed 4 words
- Ranked randomly

- Queries composed 3 words
- Ranked randomly
Baseline 2: TF-IDF

- Queries are ranked by the cosine similarity to the original query in the descending order.
- Ex: \( q = \) "a boy makes paints seeds"

<table>
<thead>
<tr>
<th></th>
<th>a boy</th>
<th>makes</th>
<th>paints</th>
<th>blue</th>
<th>seeds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Subset</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Experiment

• Generate queries with data mentioned in the preliminary experiment
  • Question and answer pairs collected in the Japanese QA site
• Use the NDL search engine to execute generated queries
• Compare results by Mean Reciprocal Rank (MRR) and distribution of the rank of target books
• Number of Words method has a random factor
  • the average of 10 runs
Result (1)

• MRR

<table>
<thead>
<tr>
<th></th>
<th>PRP</th>
<th>Number of Words</th>
<th>TF-IDF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.157</td>
<td>0.164</td>
<td>0.121</td>
</tr>
</tbody>
</table>

→ PRP was
• better than TF-IDF
• worse than Number of Words method
Result (2)

• Distribution of the rank of target books

- PRP
- Number of Words
- TF-IDF

Some queries ranked target books very low

→ PRP is more stable than Number of Words method
Discussion

• A query has more words
  • the probability that the correct description includes all query words becomes low
  • The result of this query may not contain the target book

• A query has less words
  • The number of search results becomes larger
  • The average rank of the target book becomes lower

→ PRP can control this trade-off appropriately
Conclusion

• Our method consists of two parts
  1. Ranking subsets of the original sentence query
     • Classify words in a subset query into 4 semantic roles
     • Use reliability of each query word for ranking queries
  2. Concatenating the results of subset queries

• Conduct an experiment to compare ranking methods
  • PRP was better than TF-IDF in MRR
  • PRP was more stable than Number of Words method