English Subtopic Mining

External Resource Based Subtopic Mining

- Query completion
- Query suggestion
- Google Insights
- Google keywords generator
- Disambiguation pages

Snippets of top 50 search results

Hierarchical clustering

- Subtopics are normalized and linearly combined.
- Candidate subtopics

Top Results Based Subtopic Mining

- TMiner search results
- Extract contents with “h1” tag
- Extract in-link anchor texts
- Extract the result snippets

Fragments presented by VSM

PAM clustering algorithm

- Cluster name generation

- PAM clustering algorithm

Randomly select k centers

Associate each data point to the closest center.

Swap data points between each center and non-center data point pair to find the lowest cost.

Stop the algorithm when there is no changes.

Subtopics mined in these two ways are linearly combined.

The duplicated subtopics are removed according to the WordNet-based semantic similarity.

Chinese Subtopic Mining

Extract Candidate Subtopics

- Search engines
- Wikipedia and Hudong encyclopedia
- Query suggestion
- Disambiguation pages
- Summaries of the topic page

Subtopics voted by the sources and re-ranked by the weight

- \[ \text{weight}_{\text{new}} = \text{votes} + 0.05 \times (\text{average rate}) + 0.005/(\text{intert length}) \]

- We combine the title and the snippet of the top 10 search results to form a snippet document and give every term in this document a score.

- \[ \text{TermScore}(t_i) = \frac{1}{n} \sum_{q \in Q} \frac{\text{freq}_{\text{snippet}}(t_i \times q) \times \text{freq}_{\text{snippet}}(t_i \times CT_i)}{\text{freq}_{\text{snippet}}(t_i \times q) + \text{freq}_{\text{snippet}}(t_i \times CT_i)} \]

- Only the top k terms in the snippet document are considered, and the term score are normalized.

- \[ \text{NormScore}(t_i) = 1.0 - (a - \beta) \times \frac{\text{SnippetScore}(s, k)}{\text{SnippetScore}(s, k) + (1 - \lambda) \times \text{SnippetScore}(s, k)} \]

LDA on Snippet Click Document

- Remove all the appearances of given query \( d \) from, and get a new document \( d' \).
- Estimate the latent topics \( t_1, t_2, ..., t_k \) of \( d' \).
- Get two words with the largest probabilities within each topic, denoted by \( w_{t_1} \) and \( w_{t_2} \).
- Connect up \( q \) to \( w_{t_1} \) and \( w_{t_2} \), and get 4 different phrases.
- If any of the phrases has appeared in the snippet click document \( d \), add the phrase into the intent candidate list with weight 0.4.

Document Ranking

Selective Diversification

- We only diversify the search result when a query is informational.
- To identify whether a query is informational or navigational, we leverage C4.5 algorithm to learn a decision tree.
- The features used in this algorithm are as follows:

  \[ \text{match}(\text{Sessions of } q \text{ that involves less than } n \text{ clicks}) \times (\text{session of } q) \]

  \[ \text{match}(\text{Sessions of } q \text{ that involves clicks only on top } n \text{ results}) \times (\text{Session of } q) \]

  \[ \text{CD}(@) \times (\text{Click on the most popular result of } q) \times (\text{Click on all results of } q) \]

Result Diversification Based on Novelty

- Set \( S = d_s \)
- Add \( d_s \) to \( S \)
- \[ \delta = \arg \max \delta(s, S) \]
- \[ \delta(s, S) = \sum (N - n_s) + \lambda \times \cos(w_j, w_k) \]
- End