Semantic Suggestions in Information Retrieval

Andreas Schmidt

Department of Informatics and Business Information Systems
University of Applied Sciences Karlsruhe
Germany

Institute for Applied Computer Sciences
Karlsruhe Institute of Technologie
Germany
Outlook

- Introduction
- Query Principles
- Implementation Aspects
- Summary & Outlook
STICS [1]

- Semantic Search engine for entities and categories, developed at MPI for Informatics
- Every document in the search space is preprocessed with
  - Named Entity Recognition (NER)
  - Named Entity Disambiguation (NED)
- Categories and entities based on YAGO knowledge base
- Auto-completion feature for a given prefix, based on global relevance of a entity, category (ranking)
- This often leads to empty resultsets
Problem Statement

- Given:
  - News article collection (~4 million news articles)
  - 600,000 different mentioned entities
  - 60 million occurrences of entities in collection
- Build a query interface, so that...
  - Given a number of previously chosen entities and one or more prefixes, suggest related entities, so that the result set is not empty
  - Rank the suggestions based on relevance
  - The suggestions must be calculated fast
Example

- Entity Donald Trump
  - Prefix: 'sa'
- Entity Hollywood
  - Prefix: 'sa'

Demo: https://stics.mpi-inf.mpg.de/
Query Principle

If no entity is already given:

• Extend the prefix with matching entities, ranked by their global relevance

One or more entities given:

• Select all news articles, that contain the already given entities
• Further restrict this set, by deleting all news articles not containing further entities with the given prefixes
• Extract from this set of news articles all entities with the given prefix(es)
• Rank these entities according to some relevance criteria
Ranking of Entities

- Based on some information extracted from YAGO knowledge (Wiki links)
  - Milne Witten [5]
  - Kore [6]
  - ...
- Based on „dynamic document frequency“ (in how many news documents of the resulting news document does the entity appear)
- Based on co-occurrence of entities in an interval of words inside documents
Calculation of relatedness based on co-occurrence in news

• Search for tuples, triples, quadruples of entities in the news text
• Entities in tuples must be inside an interval of length $d_{\text{max}}$ (a priori fixed)

$$d_{\text{max}} = \max(d_1, d_2, d_3)$$

• relatedness $(e_1, e_2, e_3) = \log_2(1/d_1) + \log_2(1/d_2) + \log_2(1/d_3) + ...$
Calculation of relatedness based on co-occurrence in news

- Because of strong time constraint ...
  - Precalculate "relatedness" for all tuples, triples, quadruples, ... of entities based on document collection (n > 1, n < 7) 
    \((e_1, \ldots, e_{n-1}) \rightarrow (e_n, \text{rel}_1, \ldots, \text{rel}_{n-1})\)

- Some numbers (based on 3,582,098 news articles)
  - 5,594,390 cooccurrence tuples (max dist.: 30)
  - 5,022,237 cooccurrence triples (max. dist. 42)
  - 2,814,076 cooccurrence quadruples (max. dist: 51)
  - 2,336,808 cooccurrence quintuples (max. dist.: 60)
  - 1,454,580 cooccurrence 6-tuples (max. dist.: 67)
Implementation based on relational DB

- Precalculation of tuples/triples of related entities together with a weight

- 2 entities (one entity already selected, second is suggestion):
  
  $e_1 \mid e_r \mid \text{weight}$

- 3 entities (two entities already selected, third is suggestion):
  
  $e_1 \mid e_2 \mid e_r \mid \text{weight}$

- ...

- Entity Frequency (if no entity is given so far)
  
  $e \mid \text{frequency}$
Prefix Handling

- Every entity has a short description of avg: 2.5 words

<table>
<thead>
<tr>
<th>entity_id</th>
<th>entity_value</th>
<th>human_readable_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>20195899</td>
<td>Boston_University_Bridge</td>
<td>Boston University Bridge</td>
</tr>
<tr>
<td>12905648</td>
<td>Boston_University_College_of_Communication</td>
<td>Boston University College of Communication</td>
</tr>
<tr>
<td>21356536</td>
<td>Boston_University_School_of_Law</td>
<td>Boston University School of Law</td>
</tr>
<tr>
<td>21181981</td>
<td>Boston_University_School_of_Management</td>
<td>Boston University School of Management</td>
</tr>
<tr>
<td>12738583</td>
<td>Boston_University_School_of_Medicine</td>
<td>Boston University School of Medicine</td>
</tr>
<tr>
<td>20146803</td>
<td>Boston_University_School_of_Public_Health</td>
<td>Boston University School of Public Health</td>
</tr>
<tr>
<td>11722953</td>
<td>Boston_University_School_of_Social_Work</td>
<td>Boston University School of Social Work</td>
</tr>
<tr>
<td>20534109</td>
<td>Boston_University_School_of_Theology</td>
<td>Boston University School of Theology</td>
</tr>
<tr>
<td>7782976</td>
<td>Boston_University_Tanglewood_Institute</td>
<td>Boston University Tanglewood Institute</td>
</tr>
<tr>
<td>20940206</td>
<td>Boston_University_Terriers</td>
<td>Boston University Terriers</td>
</tr>
<tr>
<td>19475745</td>
<td>Boston_University_Terriers_men\ uc027s_ice_h...</td>
<td>Boston University Terriers men's ice hockey</td>
</tr>
</tbody>
</table>

- Prefix match every start of a word
  i.e. prefix('bos', 'hea') -> (Boston_Public_Health_Commission, Boston_University_School_of_Public_Health)
• Table prefix_entity (~10 million entries)

<table>
<thead>
<tr>
<th>id</th>
<th>entity</th>
<th>word_match</th>
</tr>
</thead>
<tbody>
<tr>
<td>bos</td>
<td>1112035</td>
<td>0</td>
</tr>
<tr>
<td>bos</td>
<td>1885753</td>
<td>1</td>
</tr>
<tr>
<td>bos</td>
<td>2303237</td>
<td>0</td>
</tr>
<tr>
<td>bos</td>
<td>2417071</td>
<td>0</td>
</tr>
<tr>
<td>bos</td>
<td>2449991</td>
<td>0</td>
</tr>
<tr>
<td>bos</td>
<td>2801432</td>
<td>0</td>
</tr>
<tr>
<td>bos</td>
<td>3254453</td>
<td>0</td>
</tr>
<tr>
<td>bos</td>
<td>3702676</td>
<td>1</td>
</tr>
<tr>
<td>bos</td>
<td>3965776</td>
<td>0</td>
</tr>
<tr>
<td>bos</td>
<td>4476312</td>
<td>0</td>
</tr>
</tbody>
</table>

Additional ranking factors:

• Full word match
  (prefix 'frank' in „Frank Walter Steinmeier“ vs. „Frankfurt am Main“

• Overlap of prefix with words in column 'human_readable_name'
  (prefix 'us' in „USA“ vs. „Usenet“

• Number of words in 'human_readable_name' column
Extensions I

- Time travel queries
  - Queries restricted to an interval of time
  - Time point queries

- Restrict suggestions on news documents inside a given time interval (or point)

- Approach:
  - Split precalculated data into „slices“ of one month length
  - Calculation of bigger time intervals based on aggregation over month slices
Extensions II

- Beside entities, also categories can be used as query input.
- Integration of categories (also from wikipedia)
- Categories form a taxonomy
- Quantative aspects
  - ~250,000 categories
  - avg(6.3) categories/entity
Semantic of Categories in Queries

- **Input:**
  - Entities $e_1, ..., e_n$
  - Categories $c_1, ..., c_m$
  - Prefix $p$ (can be an entity or a category)

- **Output (Suggestion):**
  - **Entities** with prefix $p$ which can be found in news articles which contain
    1. each given entity ($e_1, ..., e_n$) and
    2. at least one entity of each given category ($c_1, ..., c_m$)
  - **Categories** with prefix $p$ from entities which can be found in news articles
    which contain
    1. each given entity ($e_1, ..., e_n$) and
    2. at least one entity of each given category ($c_1, ..., c_m$)
Summary

• Auto-completion system for query input
• Document set with preclassified entities (Disambiguation via AIDA)
• Input can be entities, categories and prefixes
• Entities and categories are from YAGO (base Wikipedia)
• High time constraints ( < 0.1 sec.)
• Precalculation of „relatedness“ based on document corpus
• Actual implementation based on relational database
Outlook

• Develop sophisticated data-structure to also handle higher volumes of data
• Integration of „normal“ words (not only entities and categories)
• Incremental updates of precalculated data-structure
• Integration of corpus independent knowledge
References


