Semantic Service Retrieval based on Natural Language Querying and Semantic Similarity

Richard Eckart de Castilho
Iryna Gurevych
Our Goal

Discover a web service offering a particular functionality

1. Treat technical web-service descriptions as natural language artifacts

2. Use information retrieval for service discovery

3. Use semantic methods to bridge the vocabulary gap
Examples

Queries

- ... service to get the author of the novel.

Service Description

```
book person service

book person

person

person

book book
is titled title
has type book type
written by author

person

person person
```
What is a Web Service?

- Reusable and self-describing software component
- Supports remote method invocation
- Supports web standards
  - Description: e.g. Web Service Description Language (WSDL)
  - Invocation: e.g. REST or SOAP

Examples

- Currency converter
- Search engines
- Travel reservation systems
- Online shop
- Digital library
- Spam filter
What makes IR semantic?

- Traditional IR: No match if query and document use different terms
- Semantic IR: match if query terms are similar to document terms
Knowledge-based Models

- Explicitly modeled knowledge
- Hypernym/hyponym relation

Examples
- WordNet
- EuroWordNet
- …

Diagram:
- General to specific relationships
  - thing
    - car
    - tree
    - bus
    - van
    - birch
    - pine
    - …
Semantic Similarity using Knowledge-base

- Term similarity
- Calculate similarity using path between two concepts
Aggregation of Term Similarities

- Two sets of terms A, B
- Calculate similarity for each pair of terms ($O(n^2)$)
- Various aggregation strategies
  - Max, average, median, geometric mean…
  - Bi-partite graph matching
Weighted Bi-Partite Graph Matching

- Two sets of terms A, B
- Each element of A may be associated to at most one element of B
- Finds best semantic mapping between concepts in query and document
Knowledge-based Fuzzy Set Model

- **Intention**
  - Use a knowledge-base
  - Avoid calculating pair-wise term similarities
  - Pre-calculate semantic representation of documents
  - Create semantic representation of query on-the-fly

- **Introducing: Semantic Context**
  - Fuzzy set of terms
  - Start with a base set of terms
  - Assign weight 1.0 to each term
  - Add all hypernyms of the terms
  - Assign semantic relatedness between term and its hypernyms as weight
  - Compare query and document using set similarity measure (Dice)
Intrinsic Information Content

Seco et al., 2004

- Information content (IC)
  - “Specificity” of a concept

- Intrinsic information content (IIC)
  - Number of hyponyms compared to size of knowledge base

\[ \text{IIC}(c) = 1 - \frac{\log(|h(c)| + 1)}{\log|K|} \]

|K| = 17
Semantic Context of a Term

- Hypernyms used to create the semantic context of a term
- Information content determines relatedness between term and hypernym
Semantic Context of a Set of Terms
Semantic Context of a Set of Terms

![Diagram of semantic context]

1. General
2. Specific
3. General
4. Specific

Values: 0.63, 0.86, 1, 0.32, 0.44, 1, 0.52
Semantic Context of two Sets of Terms

<table>
<thead>
<tr>
<th>Query</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 .63</td>
<td>4 .38</td>
</tr>
<tr>
<td>.86</td>
<td>.24</td>
</tr>
<tr>
<td>1</td>
<td>.48</td>
</tr>
<tr>
<td>.32</td>
<td>.70</td>
</tr>
<tr>
<td>.44</td>
<td>.82</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

max

max
Merging the Semantic Contexts

Query

1. 1
   .63

2. 2
   .86

3. 3

4. 4
   .32

5. 5
   .44

6. 6

max

Document

4

.48

5

.71

6

.85

max

Query

4

.32

3

.70

5

.82

2

1

max

Document

5

1

4

1

max

6

1
### Calculating Similarity of Semantic Contexts

- Dice coefficient to calculate similarity
- Dice generalizes to fuzzy sets
  - Intersection operator $\rightarrow \text{min}$, $|A| \rightarrow \text{sum}$

\[
\frac{2|(A \cap B)|(x)}{|A|(x) + |B|(x)} = \frac{2 \sum \min[A(x), B(x)]}{\sum A(x) + \sum B(x)}
\]

<table>
<thead>
<tr>
<th>Query</th>
<th>Doc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>4-6</td>
</tr>
<tr>
<td>.63</td>
<td>.48</td>
</tr>
<tr>
<td>.86</td>
<td>.71</td>
</tr>
<tr>
<td>1</td>
<td>.85</td>
</tr>
<tr>
<td>.32</td>
<td>.70</td>
</tr>
<tr>
<td>.44</td>
<td>.82</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>.52</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[\sum A(x) = 5.77\]
\[\sum B(x) = 6.56\]
Calculating Similarity of Semantic Contexts

- Dice coefficient to calculate similarity
- Dice generalizes to fuzzy sets
  - Intersection operator -> $\text{min}$, $|A|$ -> sum

\[
\frac{2|(A \cap B)(x)|}{|A|(x) + |B|(x)} = \frac{2 \sum \text{min}[A(x), B(x)]}{\sum A(x) + \sum B(x)}
\]

\[
\sum A(x) = 5.77 \\
\sum B(x) = 6.56 \\
2 \sum [A(x), B(X)] = 6.64 \\
\frac{6.64}{5.77 + 6.56} = 0.53 \text{ similarity score}
\]
Evaluated Models

- Baseline (stemmed)
  - Apache Lucene

- Path-based models (lemmatized)
  - Resnik (1995)
  - Jiang & Conrath (1997)
  - Lin (1998)

- Statistical (lemmatized)
  - Latent Semantic Analysis (LSA) by Landauer et al. (1998)
Evaluation Data Set

- OWLS-TC 3rev1 benchmark for semantic service retrieval
  [http://www.semwebcentral.org/frs/?group id=89](http://www.semwebcentral.org/frs/?group id=89)
  - 29 queries
  - 999 service descriptions (WSDL)
  - 3584 judgements
  - Used in Semantic Service Selection Contest

- Semantic resources
  - WordNet for Path-based models
  - WordNet for Fuzzy Set Model
  - Wiktionary for ESA
  - LSA index created on evaluation data set itself

- Evaluated on Mean Average Precision (MAP)
## Term Similarity using Weighted Bi-Partite Graph Matching for Aggregation

<table>
<thead>
<tr>
<th>Model</th>
<th>inner product</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lucene (baseline)</td>
<td>Cosine</td>
<td>.6558</td>
</tr>
<tr>
<td>Jiang &amp; Conrath</td>
<td>-</td>
<td>.4477</td>
</tr>
<tr>
<td>Lin</td>
<td>-</td>
<td>.4772</td>
</tr>
<tr>
<td>Resnik</td>
<td>-</td>
<td>.5314</td>
</tr>
<tr>
<td>Fuzzy Set Model</td>
<td>Dice</td>
<td>.6372</td>
</tr>
<tr>
<td>ESA</td>
<td>Avg. Prod.</td>
<td>.6515</td>
</tr>
<tr>
<td>LSA</td>
<td>Cosine</td>
<td>.6815</td>
</tr>
</tbody>
</table>
**Modified Dice Coefficient**

- Document is good match when all query terms are found
- Document is bad match when some query items are missed
- Dice heavily penalizes if document contains more terms than query

- Modified dice $d_{fl}$
  - Penalty if document does not contain all query terms
  - Never drops below 0.5 if document contains all query terms

\[ c = \frac{|A|(x)}{\max(|A|(x),|B|(x))} \]

\[ d_{fl}(A(x), B(x)) = \frac{(1 + c)|(A \cap B)|(x)}{2|A|(x)} \]

*Similarity score for $|A| = 10$ and $|A \cap B| = \min(|A|, |B|)*
Modified Dice Coefficient

- Document is good match when all query terms are found
- Document is bad match when some query items are missed
- Dice heavily penalizes if document contains more terms than query

- Modified dice $d_{fl}^*$
  - Penalty if document does not contain all query terms
  - Score drops slowly with document length if all query terms are present

$$m = \max(|A|(x), |B|(x))$$
$$c = \frac{|A|(x)}{m}$$
$$d_{fl}^*(A(x), B(x)) = \frac{(1 + c)(A \cap B)(x)}{2(|A|(x) + \log(m - |A|(x) + 1))}$$

Similarity score for $|A| = 10$ and $|A \cap B| = \min(|A|, |B|)$
Term Set Similarity using Vector/Set Aggregation

<table>
<thead>
<tr>
<th>Model</th>
<th>inner product</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lucene (baseline)</td>
<td>Cosine</td>
<td>.6558</td>
</tr>
<tr>
<td>ESA</td>
<td>Avg. Prod.</td>
<td>.5579</td>
</tr>
<tr>
<td>Fuzzy Set Model</td>
<td>Dice</td>
<td>.6037</td>
</tr>
<tr>
<td>LSA</td>
<td>Cosine</td>
<td>.6822</td>
</tr>
</tbody>
</table>
Combination of Semantic and Non-Semantic Models

- Use non-semantic model ($s_{bow}$) in addition to semantic model ($s_{sem}$)
  - Apache Lucene
- Calculate final score using linear combination
  \[ w \times s_{sem}(Q,D) + (1 - w) \times s_{bow}(Q,D) \]
- Parameter $w$ controls balance between semantic and non-semantic
# Term Set Similarity Combined with Non-Semantic Model

<table>
<thead>
<tr>
<th>Model</th>
<th>inner product</th>
<th>MAP</th>
<th>w</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lucene (baseline)</td>
<td>Cosine</td>
<td>.6558</td>
<td>-</td>
</tr>
<tr>
<td>ESA</td>
<td>Avg. Prod.</td>
<td>.6884</td>
<td>.8</td>
</tr>
<tr>
<td>LSA</td>
<td>Cosine</td>
<td>.7012</td>
<td>.4</td>
</tr>
<tr>
<td>Fuzzy Set Model</td>
<td>$d_f^*$</td>
<td>.7033</td>
<td>.4</td>
</tr>
</tbody>
</table>
Conclusion

- Successful queried technical service descriptions using natural language
- Vocabulary gap addressed using semantic models

- Fuzzy set model
  - Improved knowledge-based model
  - Semantic document representations can be pre-calculated
  - Outperformed path-based models
  - In combination with Lucene on par with LSA

- Fuzzy set model seems a good option if a knowledge-base is available
Outlook

- Fuzzy set model yields good results on this data set

- Re-run the experiments
  - With a different data set
  - With additional knowledge bases (e.g. ontologies)
  - Use a separate training set to determine $w$
  - More meaningful combination of knowledge-base and TF/IDF statistics
Thank you! Questions?

Thanks to all my colleagues at UKP!