Overview of a Semantic QA System

Aug. 17, 2011

Sin-Jae Kang*, In-Su Kang**
Daegu Univ.*, Kyungsung Univ.**
South Korea
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Overview</td>
</tr>
<tr>
<td>2. Hedge Detection</td>
</tr>
<tr>
<td>3. Cross-Lingual Link Discovery (CLLD)</td>
</tr>
<tr>
<td>4. RDF Generation &amp; LOD Extension</td>
</tr>
<tr>
<td>5. User Profile Generation</td>
</tr>
<tr>
<td>6. Semantic QA System</td>
</tr>
</tbody>
</table>
Overview

Development Process of Semantic QA System

- Step 1: Hedge detection (ongoing)
- Step 2: Cross-lingual link discovery (ongoing)
- Step 3: RDF generation & LOD extension
- Step 4: User profile generation (ongoing)
- Step 5: RDF graph search & answer generation
Hedge

In NLP (particularly, in IE), many applications aim at extracting factual information from text.

In order to distinguish facts from unreliable or uncertain information, linguistic devices such as hedges have to be identified.

indicating that authors do not or cannot back up their opinions/statements with facts

Examples of hedge cue words

some, may, probably, many, generally, perhaps, …

is considered, some cases, it is possible, it is believed, …
Step 1 : Hedge Detection

For English and Korean Wikipedia documents, hedge detection is applied.

Detected non-hedge sentences are used to extract **information to be added** into LOD.

---

### Training Step

1. Parsing & preprocessing
2. POS tagging & Chunking
3. Feature extraction (word, lemma, POS, chunk tag, …)
4. Machine learning for each algorithm

### Testing Step

1. Parsing & preprocessing
2. POS tagging & Chunking
3. Feature extraction (word, lemma, POS, chunk tag, …)
4. Applying ML algorithms

---

Hedge training corpus

Hedge testing corpus

CRF  SVM  k-NN  DT

Hedge results
Cross-Lingual Link Discovery (CLLD)

- Automatically finding potential links between documents in different languages
- not directly related to traditional cross-lingual information retrieval (CLIR)
- CLIR creates a virtual link between the provided cross-lingual query and the retrieved documents
- CLLD actively extracts a set of meaningful anchors in the source document and uses them as queries with the contextual information from the text to establish links with documents in other languages.
The purpose of CLLD

- break the language barrier in knowledge sharing

English-to-Korean CLLD

- Finding outgoing links starting from English source documents to Korean target documents
- Later, this can help to find out the location where Korean information should be added to DBpedia.
Step 2 : English to Korean CLLD

Wikipedia dump

- Wikipedia article titles
- Link probability
- Eng-Kor Cross-lingual links
- Eng-Kor Bilingual dictionary

controlled vocabulary

CLLD

Candidate Anchor Extraction

Anchor Ranking

Extract Contextual Information of Anchors

Word Sense Disambiguation

Target Word Selection

Text with linked anchors
Step 3: RDF generation & LOD extension

Generated RDFs are added to LOD through the synsets of WordNet and DBpedia.
Non-hedge sentences

Anchor (= keyword)

Cross-lingual links

1. Search LOD with English anchor

Entities

Retrieved RDF sub-graph
(Location to be inserted Korean information)

Non-hedge sentences

2. Generate RDF graphs

3. Generalize RDF nodes (WSD with WordNet)

4. Merge RDF graphs

5. Update RDF graphs (adding new Korean information to LOD)

Initial RDF graphs

Generalized RDF graphs

Linked Data

From Wikipedia, the free encyclopedia

This article may require cleanup to meet Wikipedia's quality standards. Please improve it or discuss

Non-hedge sentences

Anchor (= keyword)

Cross-lingual links

1. Search LOD with English anchor

Entities

Retrieved RDF sub-graph
(Location to be inserted Korean information)

Non-hedge sentences

2. Generate RDF graphs

3. Generalize RDF nodes (WSD with WordNet)

4. Merge RDF graphs

5. Update RDF graphs (adding new Korean information to LOD)

Initial RDF graphs

Generalized RDF graphs
User Profile

- Can help to implement recommendation systems or personalized systems

Social Web Sites

- YouTube, Flickr, Del.icio.us, ...
  - Contain **tags** per target object (ex. video, photo, bookmark, …)
  - Used tags reflect user’s current interest and taste
  - By analyzing & clustering tags → Generating user profiles automatically

- Twitter
  - By extracting & analyzing terms used in short sentences → Generating user profiles automatically
Step 4 : User Profile Generation

Tag-to-Object Vectors

<table>
<thead>
<tr>
<th>Obj_1</th>
<th>Obj_2</th>
<th>Obj_3</th>
<th>...</th>
<th>Obj_m</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag_n</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tagging

TagCluster 1 (TC_1)
TagCluster2 (TC_2)
... TagCluster i (TC_i)

Tagger-to-TagCluster Vectors (User Profile)

<table>
<thead>
<tr>
<th>TC_1</th>
<th>TC_2</th>
<th>TC_3</th>
<th>...</th>
<th>TC_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tagger_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tagger_2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tagger_k</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tagging

TaggerCluster 1
TaggerCluster2
... TaggerCluster j

Profile DB

User Profiles (Tagger vector)
Group Profiles (Tagger cluster)
Step 5: Semantic QA System

Hedge Detection

Non-hedge sentences

Fact sentences

CLLD

Cross-lingual links

WIKIPEDIA

The Free Encyclopedia

LOD Extension

RDF

WordNet

Upper Ontology (SUMO)

DBpedia

YAGO2

User Profile Generation

Semantic QA System

LOD Graph Search

Query Analysis

Answer Generation

answer

query
Concluding Remarks

Through these steps,

- Can achieve LOD extension with Korean information
- Can obtain a fundamental platform to implement personalized services

Need to develop several techniques

- In NLP, WSD, sentence generation, …
- Efficient search and matching algorithms on huge RDF graph, …

Long way to go …