Semantic Cloud Generation based on Linked Data for Efficient Semantic Annotation

- Korea-Germany Joint Workshop for LOD2 2011 -

Han-Gyu Ko

Dept. of Computer Science, KAIST
Table Contents

- Introduction
  - Motivation & Objective
  - Requirements for Semantic Cloud

- Semantic Annotation Scheme

- Semantic Cloud Generation Approach
  - Locating Spotting Points
  - Selecting Relations to Traverse
  - Identifying Similarity & Clustering

- A Case Study
  - Analysis of Generated Semantic Clouds
  - Implementation in the Web-based IPTV Environment

- Conclusion
Motivation

- To provide semantic annotation as an alternative method for user-generated metadata

- To overcome the problem of semantic ambiguity of tags with plain texts
  - e.g., “Apple” the fruit and “Apple” the company

- To fulfill the requirements of scalability and usability

  Problems of previous efforts on semantic annotation [5, 6]
  - Use terms from ontologies created by domain experts
  - Do not provide sufficient options to cover various kinds of semantics
  - Do not necessarily reflect newly created knowledge in an up-to-date manner

Copyright (c) Korea Advanced Institute of Science and Technology
August 16, 2011
Objectives

- To provide a **semantic-cloud-based annotation scheme**
  - Use semantic clouds as the primary interface
  - Easy to add semantic annotation in resource-constrained environments (e.g., smart phones and IPTVs)

- To propose the framework of **generating efficient semantic clouds**
  - To allow users to intuitively recognize candidate concepts with resolving semantic ambiguity
  - To utilize Linked Data to generate semantic clouds
Semantic-Cloud-based Annotation Scheme

1. Keyword
2. LOD Query Manager
3. Recommendation
4. Semantic Tagging

Recommended Semantic Clouds:
- Tag a
- Tag b
- Tag c

Web Contents:
- Web Content a
- Web Content b
- Web Content c
Linked Data is large-scale & heterogeneous Semantic Web data
- More than 2.5 billions RDFs from over 200 different datasets [14]
- The proposed approach need to be **incremental** and **iterative**

Three phases in the semantic cloud generation:
Phase 1: Locating Spotting Points

- Traversing and finding relevant RDF nodes starting from more important and densely connected RDF nodes is more efficient.

- **Basic Principles**
  - More abstract nodes have higher degree of connections.
  - Relative concept hierarchy can be browsed by using SKOS relationships (skos:broader, skos:narrower).

![Diagram showing the process of locating spot points using SKOS parsing and RDF prioritizing.](image-url)
Phase 2: Selecting Relations to Traverse

- Computational complexity for traversing RDF nodes can be reduced by **considering relevant relations only**

- Basic Principles
  - Semantically relevant relations can be found by considering **user contexts** such as interests and preferences
  - User interests interlink with **specific relation terms**

- W3C Recommendations
  - Traverse well-defined and popular terms
    - e.g.) FOAF, DC, SIOC, and SKOS
Phase 3: Identifying Similarity & Clustering

- **Semantic similarity** between RDF nodes is measured to decide whether to include a visited node into a cloud.

- **Basic Principles**
  - As the **number of common terms** becomes larger, the similarity increases.
  - As the **number of hops from the spotting point** becomes larger, the similarity decreases.

- **Similarity Formulas**

  \[
  \text{TermFreq}(l_1, l_2) = \frac{n(l_1, l_2)}{n(l_1) + n(l_1, l_2) / n(l_2)}
  \]
  
  \(l_1 \text{ and } l_2\): the labels of RDF nodes to compare
  
  \(n(l)\): the number of query responses for \(l\)
  
  \(n(l_1, l_2)\): the number of query responses for both \(l_1, l_2\)

  \[
  \text{SemSim}(l_1, l_2) = \frac{\text{TermFreq}(l_1, l_2)}{w^h}
  \]
  
  \(h\): the number of hops to traverse
  
  \(w\): a weight value
To compare our approach with the traditional approaches of using `rdf:type` and SKOS relationships.

**Data Preparation**

- RDF triples from Sindice Search API for keyword ‘apple’
  (http://sindice.com/developers/searchapi)

- Separation of terms despite there is semantic relevance
  - e.g., ‘Apple Inc.’ separates from ‘Apple I’, ‘Apple IIGS’

- Terms that contain keyword only
  - Balance of contents is not satisfactory (a cloud with only one term)

- Additional relevant terms that don’t contain keyword (Itunes, Macintosh in the pink cloud)
  - Each cloud is more richer than other results
Implementation in the Web-based IPTV Environment

Semantic Cloud

Selected Linked Data

Cloud Generation

Cloud Generation

Start Button

Annotation Timing

Keyword (User input)
Manipulating Linked Data in Dataset Level

1. Metadata Crawling
   - Category by Domain for each dataset

2. Graph Analysis
   - Degree (Hub)
   - Betweenness

3. Dataset Ranking
   - Category by Domain for each dataset
How to Obtain RDF triples from Linked Data

Using SPARQL Endpoints
- Also has problems
  - Some datasets don’t provide their endpoints (108 / 221 only)
  - Slow response time (more than 90 seconds in some cases)

Using BTC 2010 Data
- Provided by Falcon-S, Sindice, Swoogle, SWSE, and Watson using the MultiCrawler/SWSE framework
- !(BTC 2010 ⊃ Linked Data datasets) ➔ Didn’t work!

Using a Collection of Datasets
- Provide access to a set of copies of datasets via SPARQL endpoint
- Example
  - http://lod.openlinksw.com/sparql
  - http://uberblic.org

For Each Dataset

<table>
<thead>
<tr>
<th>Subject</th>
<th>Predicate</th>
<th>Object</th>
<th>Context</th>
</tr>
</thead>
</table>

- Duplicate quadruples deletion
- Namespace-based table classification
Which one covers the concepts for keyword most? (Coverage)

Keyword ‘Apple’
Examples of Ambiguous Tag from Del.icio.us

<table>
<thead>
<tr>
<th>Tags</th>
<th>Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>apple</td>
<td>mac, apple, computers, osx, technology, IT food, health, apple, nutrition, fruit, green</td>
</tr>
<tr>
<td>tiger</td>
<td>Photos, nature, animal, tiger, cute, animals</td>
</tr>
<tr>
<td></td>
<td>Sports, video, tiger, woods, golf, games</td>
</tr>
<tr>
<td>opera</td>
<td>music, art, opera, culture, design, download</td>
</tr>
<tr>
<td></td>
<td>software, browser, opera, web, tools, internet</td>
</tr>
</tbody>
</table>

Experimental Results

(a) Domain

(b) Dataset

(c) Triple
Summary

- Define the requirements for well-organized semantic clouds
  - Small number of clouds
  - Balance of contents in the cloud
  - No ambiguity among clouds

- Propose a Semantic Cloud Generation Framework based on Linked Data
  - Locating spotting points
  - Selecting relations to traverse
  - Identifying Similarity & Clustering

- Show that the proposed approach provides high quality of clouds via a case study
Conclusion

**Contributions**
- Efficient handling of large-scale Semantic Web data (Linked Data)
- Generating semantic clouds that enable users to
  - Specify semantics by using simply keywords
  - Intuitively recognize semantic options
  - Easily resolve semantic ambiguity

**Future Works**
- Empirical studies to decide followings
  - Optimal number of spotting point
  - Maximum number of hops to traverse
  - Threshold value to decide whether a RDF is merged in the same cloud
- Quantitative evaluation in terms of improved accuracy
- Discussion about graph traverse performance
- Measure the usability of the proposed approach
References


Questions?