Incorporating Heterogeneous Information for Mashup Discovery with Consistent Regularization

Yao Wan\textsuperscript{1}, Liang Chen\textsuperscript{2}, Qi Yu\textsuperscript{3}, Tingting Liang\textsuperscript{1} and Jian Wu\textsuperscript{1}

\textsuperscript{1}Zhejiang University, Hangzhou, China
\textsuperscript{2}School of Computer Science & Information Technology, RMIT, Melbourne, Australia
\textsuperscript{3}Rochester Institute of Technology, Rochester, USA

\{wanyao, cliang, liangtt, wujian2000\}@zju.edu.cn, qi.yu@rit.edu

April 21, 2016
Outline

- Introduction
  - Background
  - Motivation
  - Challenge

- Model
  - A probabilistic model
  - Modeling the heterogeneous network

- Experiments

- Conclusion and Future Work
What is web API?

- Application Programmable Interface (API) makes web programmable.
- RESTful service
- 10,634 APIs and 6,049 mashups in ProgrammableWeb until Nov. 2014
- Increase economic transactions from web browsers to API-driven interactions
What is Mashup?  

A mashup is lightweight web application created by combining capabilities from many other services

- Rapid creation (days not months)
- Reuses existing capabilities, but delivers new functions
- Request less technical skills
Motivation

Task: discover the component services - mashup
Challenges

- Conventional methods
  - just utilize the semantic information of service itself
  - ignore the connection between services

- Challenge
  - Design a rank algorithm
  - Integrate the rank algorithm with the network structure
A Probabilistic Model

document-centric probabilistic model: estimate the expertise of a candidate by summing the relevance of its associated documents

\[ p(m_i|q) = \lambda \sum_{a \in A_{m_i}} p(m_i|a)p(a|q) + (1 - \lambda)p(m_i|q) \]

\[ \propto \lambda \sum_{a \in A_{m_i}} p(m_i|a)p(q|a)p(a) + (1 - \lambda)p(q|m_i)p(m_i) \]

- \( p(mi|a) \): the probability of API a belongs to mashup mi
- \( p(q|a) \) and \( p(q|mi) \): the semantic similarities of API and mashup
- \( p(a) \) and \( p(mi) \): the quality of API a and mashup mi

\[ z = \lambda P_{MA} Q_{A} x + (1 - \lambda) Q_{M} y \]
Mashup Consistency Hypothesis:
- If two mashups share many common services with respect to a given query, then their relevance score in the queried field should be similar in some sense.

Objective function

\[
\Omega(z) = z^T (I - S_M) z + \mu \|z - z^0\|^2
\]

Subject to

\[
z^0 = \lambda P_{MA} Q_A x + (1 - \lambda) Q_M y
\]

Similarity matrix

\[
S_M = \Pi^{-1/2} W \Pi^{1/2} \quad \Pi_{ii} = \sum_j W_{ij}
\]
Optimization

\[ \Omega(z) = z^T (I - S_M) z + \mu \| z - z^0 \|^2 \]

s.t. \[ z^0 = \lambda P_{MA} Q_A x + (1 - \lambda) Q_M y \]

- Setting \[ \frac{\partial \Omega(z)}{\partial z} = 0 \]
  \[ (I - \alpha S_M) z^* = (1 - \alpha) z^0 \quad \alpha = 1/(1 + \mu) \]

Variant Graph Laplacian using as \( Sm \) is the adjacency matrix

- Since \( S_M \) is usually sparse
  \[ z^{t+1} \leftarrow \beta S_M z^t + (1 - \beta) \left[ \lambda P_{MA} Q_A x + (1 - \lambda) Q_M y \right] \]
  \[ \beta = 1/(1 + \mu), \quad z^* = z^\infty \text{ is the solution.} \]

Implementation

**Offline Stage**

- Service Database
  - $M_1$, $M_2$, $M_3$ Mashups
  - APIs

  - Heterogeneous Network Construction
    - $G_{MA}$, $G_A$

- NLP
- Network Analysis
  - Tf-idf Vector
  - Mashup-API Matrix
  - Quality Matrix

**Online Recommendation Stage**

- Input
- Query

  - Filtering
  - Similarity Calculation
  - Relevance Score Vector

  - Mashup Recommendation
    - $M_3$, $M_1$, $M_2$
Experiments

- Dataset

<table>
<thead>
<tr>
<th># of mashups</th>
<th>4,699</th>
</tr>
</thead>
<tbody>
<tr>
<td># of APIs</td>
<td>937</td>
</tr>
<tr>
<td># edges in $G_M$</td>
<td>3,760,923</td>
</tr>
<tr>
<td># edges in $G_{MA}$</td>
<td>8,127</td>
</tr>
</tbody>
</table>

- Evaluation metrics

$$P@K = \frac{\# \text{ relevant in top } K \text{ results}}{K}$$

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

$$\alpha-DCG_K = \sum_{i=1}^{K} \frac{G_i}{\log_2(i + 1)}$$

Clarke, C.L., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Bu’ttcher, S., MacKinnon, I.: Novelty and diversity in information retrieval evaluation.
The idea of binary judgment in the evaluation is: a result for a given query can be judged as relevant once its category is identical with the category of the query.
Experiments

- Experimental results

<table>
<thead>
<tr>
<th></th>
<th>P@10</th>
<th>P@20</th>
<th>P@50</th>
<th>MRR</th>
<th>α-DCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>PW</td>
<td>0.595</td>
<td>0.493</td>
<td>0.431</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MD-Sim</td>
<td>0.555</td>
<td>0.488</td>
<td>0.553</td>
<td>0.121</td>
<td>2.920</td>
</tr>
<tr>
<td>MD-Sim+ (vs MD-Sim)</td>
<td>0.575</td>
<td>0.495</td>
<td>0.534</td>
<td>0.137</td>
<td>2.916</td>
</tr>
<tr>
<td></td>
<td>+3.60%</td>
<td>+1.43%</td>
<td>-3.44%</td>
<td>+13.22%</td>
<td>-0.01%</td>
</tr>
<tr>
<td>MD-HIN (vs PW)</td>
<td>0.555</td>
<td><strong>0.53</strong></td>
<td><strong>0.537</strong></td>
<td><strong>0.160</strong></td>
<td>3.027</td>
</tr>
<tr>
<td></td>
<td>-6.72%</td>
<td>+7.51%</td>
<td>+24.59%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(vs MD-Sim)</td>
<td>0.00%</td>
<td>+8.61%</td>
<td>-2.89%</td>
<td>+32.23%</td>
<td>+3.66%</td>
</tr>
</tbody>
</table>

- Observations
  1. when the value of K is large, our approach has a great advantage over ProgrammableWeb search engine.
  2. From the perspective of ranking of results, our extended approach (MD-HIN) achieves better performance than the probabilistic approach.
  3. Among all the discovery methods, our proposed method generally achieves better performance on both metrics, indicating the effectiveness of our approach.
Parameter Analysis

Impact of lambda

- Observations

1. The value of $\lambda$ really has a significant impact on the performance of mashup discovery.

2. As $\lambda$ increases, the P@20 value increases at first, but when $\lambda$ surpasses a certain threshold, the P@20 value decreases with further increase of the value of $\lambda$.

3. This phenomenon confirms the intuition that purely using the semantic information of mashups or purely employing the semantic information of their related APIs cannot generate better performance than fusing these two factors together.
Parameter Analysis

□ Impact of beta

(a) Impact of $\beta$ on $P_{@20}$  (b) Impact of $\beta$ on $MRR$  (c) Impact of $\beta$ on $\alpha$-$DCG$

• Observations

• 1. $\beta$ has a significant impact on the precision of the discovery results.
Conclusion

- Propose an approach to improve mashup discovery by integrating the semantic information of mashups and their related APIs.
- Similarity consistency is proposed and a regularization framework is implied.
- Comprehensive experiments on a real-world dataset.

Future work

- Extend our ground truth with more queries.
- More evaluation metrics will be introduced.
Q & A

Thank you!

wanyao@zju.edu.cn
http://www.wanyao.me