

基于 Mashup 和服务语义的可组合服务推荐

革家象¹, 刘子豪², 许斌¹

¹ (清华大学 计算机科学与技术系, 北京 中国 100084)

² (北京信息科技大学 信息安全系, 北京 中国 100192)

E-mail : gerjiexiang@gmail.com

摘要: 互联网上每天有大量的 Web 服务和 Mashup 被创建。因此, 如何向用户推荐可能进行组合的服务 (可组合服务) 是一个具有挑战性的问题。现有的一些服务推荐方法存在二八定律, 即根据服务组合的历史信息来进行服务推荐的方法会导致流行的服务不断的被推荐, 所推荐的服务集中在 20% 的范围左右。本文提出了一种服务推荐方法, 基于已有的服务组合 (Mashup) 信息和服务之间的语义关联, 进行可组合服务推荐。我们首先介绍服务组合推荐的过程和基本架构, 然后描述如何预测潜在 Mashup 里的可组合服务和计算服务之间的语义关联度。通过在包含 11182 个 API 和 7568 个 Mashup 的数据集上进行的实验, 证明了我们的方法有效地扩大了 43% 的服务推荐范围。

关键词: 可组合服务; 服务推荐; 服务组合历史; 语义关联度

中图分类号: TP

文献标识码: A

文章编号: 1000-1220 (2014) 02--

Novel Composable Service Recommendation based on Mashup and Service Semantics

GE Jia-xiang¹, LIU Zi-Hao², XU Bin¹

¹ (Department of Computer Science & Technology, Tsinghua University, Beijing 100084, China)

² (Department of Information Security, Beijing Information Science & Technology University, City 100192, China)

Abstract: A large number of new Web service and Mashup have been created on the Internet every day. Thus, recommending services which can be composed together (Composable Service) is interesting but remains a tough problem. Current service recommendation approaches hold Pareto principle, namely recommendation rely on historic services composition information which results in popular services being recommended frequently and the recommended services are concentrated in around 20% of scale. In this paper, we proposed a novel composable recommendation based on two dimensions. These two dimensions involved current composition information and services semantic relatedness calculation. First, we presented service composition recommendation procedures and structure. Then, we describe composable service prediction in Mashup and services semantic relatedness calculation approaches. After experiments on dataset with 11182 APIs and 7568 Mashups, the result shows our approaches effectively extend 43% of recommendation scale.

Key words: Composable Service; Service Recommendation; Historic Services Composition; Semantic Relatedness

1 引言

据统计, 在 ProgrammableWeb 网站中每天都会产生数十个新的 Mashup 和被称之为 Web 服务的 API。除此之外, 还有数十个 Web 服务虽然被创建但是没有被发现或是在 UDDI 注册登记。与此同时也有数十个 Web 服务渐渐地消失, 在这样的迅速发展和更替的情况下, 找到最能符合用户需求的 Web 服务是一个迫切需要解决的问题。用户通过服务发现找到所需要的候选服务, 而服务推荐则能帮助开发者挑选最合适的服务。目前常用的仅根据服务基本功能和服务质量进行的服务推荐只能满足用户较为单一的需求。为了满足用户

更为复杂的需求, 需要进行可组合服务的推荐。

可组合服务推荐即推荐可能进行组合的服务。一般服务组合问题的研究依据 QoS、语义和本体[1][2][3]等方面。由于 QoS 信息的获取受到许多阻碍和许多服务的描述信息不够全面, 直接影响了服务推荐的准确性。近年来, 随着 Web 2.0 的发展, Mashup 也被广泛的开发和和使用, 而 Mashup 也被视为 Web 2.0 的特性之一。Mashup 组合了多个服务为一体, 形成功能更强大的服务。据此, 很多研究根据现有的 Mashup 作为服务的组合历史信息来研究服务推荐。但是, 有研究表明在 Mashup 里仅有 22.2% 的服务 (API) 被调用[6]。

收稿日期: 2014-09-15 基金项目: 本课题得到国家科技计划项目“八六三”(No. 2013AA01A607)、国家自然科学基金(No.61170212)和北京多媒体网络化重点实验室资助。作者简介: 革家象, 男, 1988 年生, 硕士研究生, 主要研究领域为在网络环境下的知识工程, 包含服务计算, 语义 Web, 社会网络分析与挖掘。刘子豪, 男, 1995 年生, 本科学生, 主要研究领域为服务计算、网络安全。许斌, 男, 1973 年生, 副教授, 博士生导师, 主要研究领域为 Web 服务计算、语义 Web 和移动计算。

因此, 根据 Mashup 的组合信息进行服务推荐, 会使流行的服务不断被推荐, 冷门的服务逐渐消失, 最后形成局部范围内的推荐, 即存在二八定律。图 1 显示少量的 APIs 才会被 Mashup 调用, 如果根据 Mashup 作为组合的历史信息来进行推荐, 大部分的 APIs 则不会被发现 (图中圆圈的颜色用来区分服务的类型, 大小可以用来表示服务操作的数量, 在此没有其他特别的意思)。

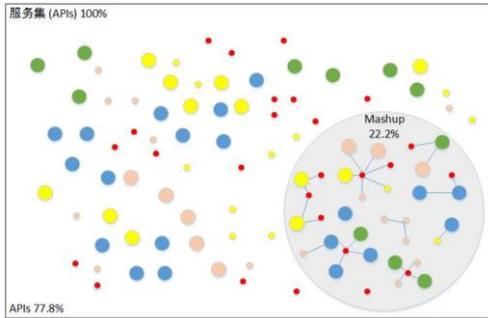


图 1 Web 服务 (APIs)

Figure 1 Web Services (APIs)

本文提出可组合服务推荐。根据现有的 Mashup 组合信息进行分析, 预测用户在服务组合上的组合习惯, 作为可组合的服务推荐。除此之外, 我们还根据服务的语义关联度来完善可组合服务的推荐。服务的基本功能描述 (Description) 和类型 (Category) 信息, 可以用来计算服务之间在语义上的关联, 而语义关联也常用来计算服务的匹配度 [7]。所以可组合服务推荐是结合现有服务的组合习惯和语义关联。

本试验的数据集涵盖 11182 个 API 和 7568 个 Mashup。通过试验我们发现可组合服务的推荐不仅能推荐出受欢迎的服务, 也能推荐出没有被调用过, 但是具有替代关系的服务。本文主要贡献如下:

- 提出一种可组合服务的推荐方法: 推荐可能进行组合的服务, 用户可以根据推荐的服务, 按自己的需要进行组合。这些可组合的服务具有协作和替代关系。
- 有效的利用现有的服务组合和语义关联, 扩大了服务推荐范围: 现有的服务组合代表用户的组合习惯, 语义关联代表服务进行组合的关联性, 两个结合一体能更有效预测出可组合的服务。通过实验证明本方法有效地扩大了 43% 的服务推荐范围。

本文的结构如下: 第二节讲述服务组合推荐的相关工作; 第三节介绍服务组合推荐过程和基本结构; 第四节介绍预测潜在 Mashup 里的可组合服务; 第五节介绍计算服务语义关联度的方法; 第六节展示实验结果和数据集; 第七节是本文总结和未来的展望。

2 相关工作

国内外有许多关于服务推荐的研究, 研究的依据大多

是用户调用服务的历史信息和服务本身的 QoS 信息。推荐系统中的“协同过滤”技术也经常被运用在服务推荐中, 所以我们将在这一节讨论协同过滤在服务推荐中的相关应用。

根据服务的历史信息进行的推荐, 首先需要收集大量用户调用服务的信息, 综合用户特征, 并进行用户分类, 然后通过相似的用户 (user-based), 或是相似的服务 (item-based), 来预测用户感兴趣的服务, 以达到用户个性化需求和潜在需求的推荐。例如结合用户和服务的潜在 (latent) 和外在 (external) 的特征的统一协同过滤模型研究的服务推荐 [7]。利用服务聚类, 综合用户特征和用户评价相似性进行用户情境聚类, 并据此对初步结果进行过滤, 实现了满足用户个性化需求的优化推荐 [8]。然而, 这些方法的重大缺陷就是, 受欢迎的服务不断的被推荐, 如果服务没有被调用过或是用户没有调用服务的历史信息, 则不能进行推荐。

基于 QoS 的服务推荐通过收集大量的用户调用服务时的 QoS 信息, 譬如响应时间和失败率, 使用协同过滤技术来预测 QoS 值并进行服务推荐 [5]。通过感知用户上下文, 使用层次分析法获得用户 QoS 偏好, 综合评估备用服务的 QoS 性能, 给出最适合当前用户的服务 [10]。然而, 根据 QoS 来进行服务推荐的最大问题是要获取大量的 QoS 信息, 而且 QoS 的准确性和调用的地理位置有关, 所以获取 QoS 信息是一件非常困难的事情。

3 服务组合的推荐过程和基本结构

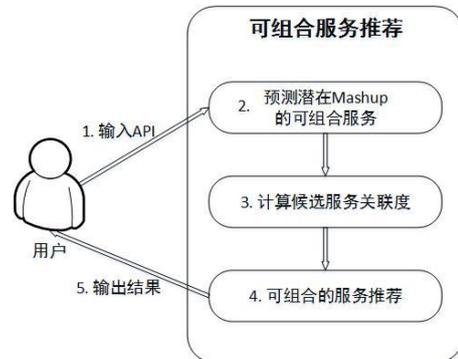


图 2 可组合服务推荐过程

Figure 2 Composable Service Recommendation Procedure

在 Web 服务中, Mashup 是指结合网络上多个服务为一体, 形成的功能更强大的服务, 所以 Mashup 是服务组合的一种。本文根据现有的 Mashup 去发现用户在服务组合上的习惯, 预测出可组合的服务, 结合服务描述的信息关联, 提出可组合服务推荐。图 2 展示可组合服务的推荐过程, 分为下列 5 个步骤:

1. 用户提供想要寻找组合对象的 API。在这个步骤假设用户的输入为 API 的名称。
2. 根据用户输入的 API 名称, 去发现调用 API 的

Mashup, 并且进行协同过滤, 预测出可能与该 API 进行组合的候选服务。

3. 再根据这些候选服务描述的信息计算服务语义关联度, 寻找出更多服务作为推荐的候选服务。
4. 通过过滤条件挑选出可组合的服务作为推荐。
5. 最后, 把推荐的结果输出给用户。

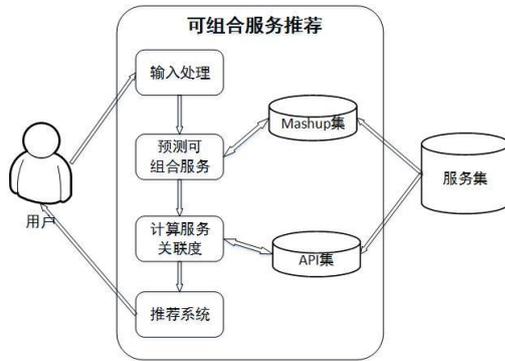


图 3 可组合服务推荐架构

Figure 3 Composable Service Recommendation Architecture

图 3 展示服务组合的推荐架构, 包含下列几个组成部分:

- 输入处理: 有时用户不知道什么样的服务才能满足用户的需求, 不知道要怎样寻找服务组合的对象。此时, 使用者可能输入一段描述, 而输入处理负责根据用户的需求匹配到准确的服务。
- 预测可组合服务: 根据输入处理得到的服务, 在 Mashup 集里面发现调用此服务的 Mashup, 通过协同过滤技术里的方法, 预测出更多可组合的服务。
- 计算服务关联度: 再根据可组合的服务和 API 集里的服务计算服务之间的关联度, 得出更多可替代关系的服务。
- 推荐系统: 综合在 Mashup 集预测出协作关系的服务, 结合 API 集发现可替代关系的服务, 进行条件过滤, 把具有代表的可组合服务推荐给用户。

4 预测可组合的服务

由于 Mashup 是不同用户开发出来的一种服务组合, 它具有其他用户的服务组合信息。通过收集这些信息, 去发现用户在服务组合上的习惯和风格, 可以预测出可组合的服务。

协同过滤技术被大量的使用到服务推荐里, 透过收集大量用户 (users) 的偏好或兴趣 (items) 信息, 来进行预测用户的兴趣。本试验把 User 视为 Mashup, 把 Item 视为 Mashup 里组合的 API, M 个 Mashup 和 N 个 API 的关系表示为 $M \times N$ 矩阵, 称为 user-item 矩阵。而每项 (entry) 值 $r_{x,i}$ 表示 Mashup x 调用 API i 的值 (有调用则为 1, 否则为 0)。然后使用 User-based 方法, 本文将称之为 Mashup-based 方法。

Mashup	调用的 APIs					
Geo Location Atlas	Yahoo Weather	Vine	Twitter	Quova	Nokia Maps	...
Whaddado	Yelp	YouTube	Yahoo Weather	Yahoo Geocoding	Flickr	...
Local Genius	Yahoo Geocoding	Yahoo Geo Location	Google Places	Google Maps		
Can I Leave?	Yahoo Weather	Yahoo Query Language	Yahoo Geo Location			
RueFind	Yahoo Weather	Yahoo Query Language	Google Maps			
Mood Music	Yahoo Weather	Last.fm	GeoNames			

图 4 Mashup-based 之协同过滤示意图

Figure 4 Mashup-based Collaborative Filtering

基于 Mashup-based 方法, 使用协同过滤技术, 即计算两个 Mashup 的相似度, 然后预测出 Mashup 之间是否存在对彼此的 API 感兴趣。图 4 展示通过预测后可以把 Google Maps 推荐给 Can I Leave?, 亦可以把 Yahoo Geo Location 推荐给 RueFind, 因此 Google Maps 和 Yahoo Geo Location 都与 Yahoo Weather 和 Yahoo Query Language 形成可以进行组合的服务, 所以本文称之为可组合服务的推荐。(附录 A 展示图里 Mashup 的基本描述)

4.1 Pearson 关联系数

Pearson 关联系数 (Pearson Correlation Coefficient: PCC) 用在推荐系统里的相似度计算。在 Web 服务里以 Mashup-based 的协同过滤方法, PCC 被用来计算调用相同 API 的 Mashup x 和 y 的相似度 (1)。

$$sim(x, y) = \frac{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r}_x)(r_{y,i} - \bar{r}_y)}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r}_x)^2 \sum_{i \in I_{xy}} (r_{y,i} - \bar{r}_y)^2}} \quad (1)$$

当 $I_{xy} = I_x \cap I_y$ 是 Mashup x 和 y 调用的 API 的子集合, $r_{x,i}$ 是 Mashup x 调用的 API i 的值 (有调用则为 1, 否则为 0), 而 \bar{r}_x 和 \bar{r}_y 分别表示是 Mashup x 和 y 所有调用的 API 的平均值。两个 Mashup x, y 的相似度 $Sim(x, y)$ 是介于 -1 和 1 之间, 其值约大相似度就越高。当两个 Mashup x, y 都没有调用共同的 API ($I_{xy} = null$), 而 $Sim(x, y)$ 为 $null$ 。

然而根据统计, 只有 1.4% 的 Mashup 调用超过 10 个 API, 平均每个 Mashup 调用 2 个以上的 API, 最少为 1 个 API, 而最多则为 65 个 API (见表 2、图 5)。据此, 如果 Mashup x 调用少量的 API 与调用多量 API 的 Mashup y , 进行相似度计算, 就会导致有一定的相似度存在, 但是实际上 Mashup x 与 y 并没有相似性。为了避免过多依赖 PCC 的相似度计算, 提出重要权重 (Significance Weighting) 来降低调用少量共同 API 的影响, 以改善 PCC 的相似度计算 [5]。

$$Sim'(x, y) = \frac{2 \times |I_x \cap I_y|}{|I_x| + |I_y|} Sim(x, y) \quad (2)$$

公式 (2) 是改善 PCC 的相似度计算, $|I_x \cap I_y|$ 是 Mashup x 和 y 调用相同 API 的数量, 而 $|I_x|$ 和 $|I_y|$ 分别是 Mashup x 和 y 各别调用 API 的数量。当 Mashup x 和 y

调用相同 API 的数量 $|I_x \cap I_y|$ 低时, 重要权重 $\frac{2 \times |I_x \cap I_y|}{|I_x| + |I_y|}$

就会降低 Mashup x 和 y 的相似度。由于 $\frac{2 \times |I_x \cap I_y|}{|I_x| + |I_y|}$ 的值

是介于 0 和 1 之间, 而 $Sim(x, y)$ 的值是介于 -1 和 1 之间, 所以 $Sim'(x, y)$ 的值介于 -1 和 1 之间。

5 服务语义关联度计算

服务组合的推荐, 是为了能推荐出更多具有替代关系的服务, 除了在 Mashup 数据集中运用协同过滤进行预测以能推荐出协作 (Collaborative) 关系的服务外, 本试验还计算了服务之间的语义相似度。我们根据服务的类型 (Category) 信息, 来计算类型上的关联度, 然后从服务描述信息提取相应的标签 (Tag) 信息, 用来计算更精准的服务关联度。下面介绍明确语义分析 (Explicit Semantic Analysis) 的方法。

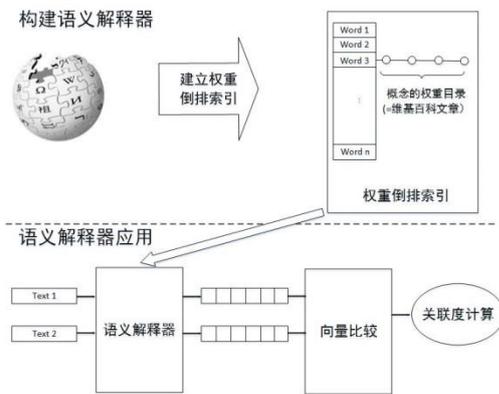


图 5 语义解释器

Figure 5 Semantic Interpreter

5.1 明确语义分析

明确语义分析的特点是充分利用了知识密集型的维基百科信息, 在语义关联度和文本分类等自然语言处理中取得了极为出色的效果 [11] [12]。明确语义分析的核心思想是建立基础概念的集合并视为一个典型的 n 维语义空间, 文本被视为该空间内的一个点, 通过计算两个点之间的距离来表示两个文本片段的语义相关度。

明确语义分析所使用的知识库需要满足以下条件:

1. 覆盖面足够宽
2. 持续的维护和更新
3. 概念需要容易被理解和应用
4. 每个概念都要和文本对应

本实验使用 2012 年的维基百科数据集, 在维基百科¹中收集的英文文章总数为四百万篇左右。

5.1.1 明确语义分析操作过程

构建语义解释器 (Interpreter)

在明确语义分析里, 一个明确定义的词可以视为一个概念。对于给定的概念集合 C_1, \dots, C_n , 和一组相关的文档 d_1, \dots, d_n , 我们建立一个每 n 列对应一个概念并且每一行对应一个出现在 $U_{i=1..n} d_i$ 中的单词的矩阵 T 。 $T[i, j]$ 就是文档 d_j 中的术语 t_i 的 TF-IDF 值 (3)。

$$T[i, j] = tf(t_i, d_j) \cdot \log \frac{n}{df_i} \quad (3)$$

其中词频 (Term Frequency) 的定义是 (4)

$$tf(t_i, d_j) = \begin{cases} 1 + \log count(t_i, d_j), & \text{if } count(t_i, d_j) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$df_i = |\{d_k : t_i \in d_k\}|$ 是包含术语 t_i 集合的文档数目。 (Document Frequency) 最后, 对每一行进行余弦标准化以消除文档长度不同带来的差别。

$$T[i, j] \leftarrow \frac{T[i, j]}{\sqrt{\sum_{l=1}^n T[i, l]^2}} \quad (5)$$

其中 r 是术语的数目。单词 t_i 的语义解释器就是矩阵 T 中的 i 行。这就是说, 一个单词的意思由一组关联由 TF-IDF 值的概念向量表示的, 这也反映着这些概念在现实生活中的相关性。矩阵 T 也可以被视为是一个倒排索引, 把每一个单词都映射到一个它出现过的概念列表。倒排索引非常有利于计算单词的解释向量之间的距离。

图 4 展示了 ESA 算法的过程。首先构建一个基于维基百科的语义解释器, 用来把文本解释为概念向量空间中的一个解释向量。计算两段文本的语义相关度时, 分别计算其解释向量, 然后计算两个向量的余弦相似度即可获得它们的语义相关度。

语义相关度计算

明确语义分析把文本表示为一个在高维度概念空间中的解释向量。根据这种表示方式, 计算文本之间的语义相关度简化为比较他们的向量。向量可以使用多种度量方式进行比较, 在明确语义分析里使用余弦度量, 所以明确语义关联度的值是 0 到 1。

6 服务组合推荐的实验

6.1 数据集

本实验于 2014 年 5 月, 从 ProgrammableWeb 网站获取 Mashup 和 API。表 2 展示在所有有效的 API 里仅有 14% 被 Mashup 调用, 这说明 API 的增长数量要比 Mashup 来得快, 而有效的 API 也表明部分的 API 逐渐的消失。图 5 展示 Mashup 调用 API 的数量, 有 52% 的 Mashup 仅调用了 1 个 API, 24% 的 Mashup 调用了 2 个 API, 和 11% 的 Mashup 调用了 3 个 API。

¹ 维基百科, http://en.wikipedia.org/wiki/Main_Page

表 1 数据集
Table 1 Data Set

Mashup 数量	7568
API 数量	11182
Mashup 调用的 API 数量	1366 (=14%)
有效的 API 数量	9654
每个 Mashup 平均调用 API 数量	2.12

6.2 服务组合推荐的实验

基于数据集，我们应用协同过滤技术，从 Mashup 数据集里，预测出具有协作关系的服务，并根据这些服务计算语义关联度，能关联出具有替代关系的服务。通过过滤条件，挑选出若干个可组合的服务作为推荐，这些服务具有协作关系和替代关系，用户可以根据之间的需求自由组合。通过试验发现，根据用户的不同需求，除了在 14%范围内的推荐外，我们的方法能扩大平均 43%的推荐范围，有效突破了二八定律的服务推荐范围限制。

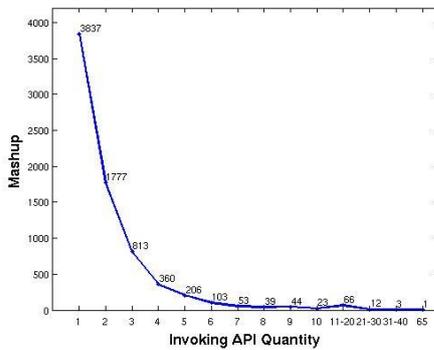


图 6 Mashup 调用 API 数量统计

Figure 6 Mashup Invoke API Statistic

图 6 为可组合服务推荐的示意图。我们假设用户的输入为 Travel，经过输入处理后，我们选择 Yahoo Travel 为用户推荐 Travel 的可组合服务，我们在 Mashup 里预测出能与 Yahoo Travel 的进行组的 API，并根据这些 API 关联出 Mashup 调用以外的 API。（图中圆圈的颜色用来区分服务的类型，大小可以用来表示服务操作的数量，在此没有其他特别的意思，附录 A 展示图里 API 的基本描述，更多的实验结果展示于 <http://sminer.org/wsrec.html>）

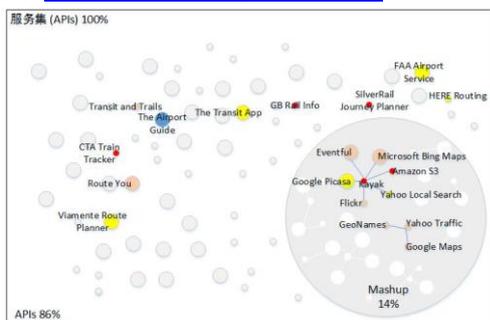


图 7 可组合服务推荐

Figure 7 Composable Service Recommendation

7 结论与展望

在服务计算研究里，服务推荐是服务选择的延伸问题，服务推荐能解决服务选择的选择最优服务问题。例如 QoS 感知的服务推荐方法[5]可以有效应用到 QoS 驱动的服务选择[13]和 QoS 驱动服务组合里的容错策略(Fault Tolerance strategy)。本实验提出一种可组合服务推荐方法，可以推荐出具有协作关系、替代关系的服务。我们根据 Mashup 调用 API 的历史信息，通过协同过滤技术，去发现现有的用户在服务组合的习惯，预测出具有协作关系的服务，用户可能根据这些服务，根据用户的需求随意组合。

由于 Mashup 里被调用的 API 数量不多，因为被调用的 API 属于流行和发展悠久的 API，所以不断被推荐和调用。为了解决服务推荐存在二八定律的问题，本实验想着于去发现潜在 Mashup 调用以外的 API，所以我们根据推荐候选协作关系的服务，计算服务关联度，推荐出更多潜在可能进行组合的服务。所以可组合服务推荐是根据用户的需求，既能推荐出其他用户习惯性组合的服务即流行的服务，也能推荐出其他具有替代性的服务，还能使推荐范围扩大，不局限于局部范围内的推荐。在未来的工作里，我们将服务组合的推荐模型应用到服务搜索系统 SMiner2 里。我们将其改进为具有服务组合推荐能力的服务搜索系统

References:

- [5] Zheng, Z., Ma, H., Lyu, M. R., & King, I. (2011). Qos-aware web service recommendation by collaborative filtering. *Services Computing, IEEE Transactions on*, 4(2), 140–152.
- [6] Huang, K., Fan, Y., & Tan, W. (2012, June). An empirical study of programmable web: a network analysis on a service-mashup system. In *Web Services (ICWS), 2012 IEEE 19th International Conference on* (pp. 552–559). IEEE.
- [7] Zhong, J., Li, A., Deng, P., & Su, L. (2012). Web Service Recommendation Based on Latent Features. *Journal of Information and Computational Science*, 9(1), 121–134.
- [9] Chen, X., Zheng, Z., Liu, X., Huang, Z., & Sun, H. (2013). Personalized qos-aware web service recommendation and visualization. *Services Computing, IEEE Transactions on*, 6(1), 35–47.
- [11] Egozi, O., Markovitch, S., & Gabrilovich, E. (2011). Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems (TOIS)*, 29(2), 8.
- [13] Zheng, Z., & Lyu, M. R. (2009, June). A QoS-aware fault tolerant middleware for dependable service composition. In *Dependable Systems & Networks, 2009.*

DSN'09. IEEE/IFIP International Conference on (pp. 239-248). IEEE.

附中文参考文献:

[1] Feng Xingjie, Wang hui, Xu Yajuan. Research on QoS-aware Web Services Composition based on Markov Decision Process. International Conference on Power Electronics and Intelligent Transportation System, Guangdong Shenzheng, China, 2010, 3(5): 346-350 (in Chinese)

(冯兴杰, 王辉, 许亚娟. 基于马氏决策过程 QoS 感知的 Web 服务组合研究. PEITS, 中国广东深圳, 2010, 3(5): 346-350)

[2] Tang Haiyan, Chen Ningjiang. A Matching Method for Semantic Web Services Based on Association Matching Degree Calculation. Journal of Guangxi Academy of Sciences, 2012, (1): 23-27 (in Chinese)

(唐海燕, 陈宁江. 一种基于关联匹配度计算的语义 Web 服务匹配方法. 广西科学院学报, 2012, (1): 23-27)

[3] Zhao Yun, Si Huayou. Automatic Composition of Service Driven by Request Based on OWL-S. CACIS, Anhui Hefei, China, 2008, 19(2): 1100-1104 (in Chinese)

(赵芸, 司华友. 基于 OWL_S 需求驱动的 WEB 服务组合研究. 计算机技术与应用进展, 中国安徽合肥, 2008, 19(1): 1100-1104)

[4] Zhang Yanxia. Web Service Matchmaking based on Semantic Similarity. Taiyuan University of Science and Technology, Technology of Computer Application, 2012. (张艳霞. 基于语义相似度的 WEB 服务匹配方法研究. 太原科技大学, 计算机应用技术, 2012.)

[8] Gu Linglan. Web Service Recommendation Method based on Context. Computer Engineering and Design, 2014, 35(3): 1115-1121 (in Chinese)

(古凌岚. 基于情境的 Web 服务推荐方法. 计算机工程与设计, 2014, 35(3): 1115-1121)

[10] Tao Chunhua, Feng Zhiyong. Novel QoS-aware Web Service Recommendation Model. Application Research of Computers, 2010, 27(10): 3902-3917 (in Chinese)

(陶春华, 冯志勇. QoS 感知的 Web 服务推荐模型. 计算机应用研究, 2010, 27(10): 3902-3917)

[12] Chen Yan, Long Jianxun. Automatic Abstraction Algorithm based on Explicit Semantic Analysis. Computer Engineering, 2011, 37(3): 183-185 (in Chinese)

(陈燕, 龙建勋. 基于明确语义分析的自动文摘算法. 计算机工程, 2011, 37(3): 183-185)

附录 A.

APIs	基本描述
Yahoo Travel	Online travel services
Yahoo Traffic	Traffic data and routing
Eventful	Local events discovery and demand

Google Maps	Mapping services
Google Picasa	Photo management and sharing service
Flickr	Photo sharing service
Yahoo Local Search	Local search service
GeoNames	Geographic name and postal code lookup
Amazon S3	Online storage services
Microsoft Bing Maps	Mapping services
Kayak	Travel search service
Transit and Trails	Bay Area outdoor excursion planner
The Airport Guide	Airport information service
CTA Train Tracker	Chicago transit tracking service
Route You	Transportation routing service
Viamante Route Planner	Italian route planning and mapping service
The Transit App	Transportation directions and routes application
GB Rail Info	British railway journey planner
SilverRail Journey Planner	Trip planning service
FAA Airport Service	Airport activity and weather information service
Here Routing	Pedestrian, public transit, and vehicle routing

Mashup	基本描述
Geo Location Atlas	Once visited, you can see nice OVI map of place based on ISP of yours with clickable foursquare places to explore.
Whaddado	Whaddado is a Social bookmarking site for events and things to do Whaddado, add your fav websites, create lists and share them with friends. Discover new things to do and follow our experts to find out the coolest things around the world.
Local Geonius	Web app that finds places of interest near your location.
Can I Leave?	Simple website that tells if you can leave your home safely. Obviously, it's about weather.
RueFind	RueFind is a travel application which tracks interesting tourist attractions around the world. Users can add, rank or create lists of their favorite attractions.
Mood Music	A site which detects your location, queries Yahoo! Weather for that location and plays the appropriate music.