

# 基于语义的关键词提取算法<sup>\*</sup>)

方俊 郭雷 王晓东

(西北工业大学自动化学院 西安 710072)

**摘要** 关键词<sup>1</sup>提供了文档内容的概要信息,它们被使用在很多数据挖掘的应用中。在目前的关键词提取算法中,我们发现词汇层面(代表意思的词)和概念层面(意思本身)的差别导致了关键字提取的不准确,比如不同语法的词可能有着相同的意思,而相同语法的词在不同的上下文有着不同的意思。为了解决这个问题,这篇文章提出使用词义代替词并且通过考虑关键候选词的语义信息来提高关键词提取算法性能的方法。与现有的关键词提取方法不同,该方法首先通过使用消歧算法,通过上下文得到候选词的词义;然后在后面的词合并、特征提取和评估的步骤中,候选词义之间的语义相关度被用来提高算法的性能。在评估算法时,我们采用一种更为有效的基于语义的评估方法与著名的Kea系统作比较。在不同领域间的实验中可以发现,当考虑语义信息后,关键词提取算法的性能能够得到很大的提高。在同领域的实验中,我们的算法的性能与Kea++算法的相近。我们的算法没有领域的限制性,因此具有更好的应用前景。

**关键词** 关键词提取,语义相关度,消歧

## Semantically Improved Automatic Keyphrase Extraction

FANG Jun GUO Lei WANG Xiao-dong

(College of Automation, Northwestern Polytechnical University, Xi'an 710072, China)

**Abstract** Keyphrases provide semantic metadata producing an overview of the content of a document, they are used in many text-mining applications. In the process of keyphrases generation, we notice that the distinction between lexical level (term for meaning) and conceptual level (the meaning itself) can result in inaccuracy. In order to solve this problem, this paper proposes a new method that improves automatic keyphrase extraction by using semantic information of candidate keyphrases. Our keyphrases extraction method, in contrast to current methods, outputs the senses set instead of terms set by using word sense disambiguation method, as sense has only one unique meaning. Semantic relatedness between senses of candidate keyphrases is taken into consideration in the stage of term conflation, feature calculation, and evaluation. We evaluate our semantically improved method against the well known Kea system by using a more effective semantically enhanced evaluation method. The inter-domain experiment shows that quality of keyphrases extraction can be improved significantly when semantic information is exploited. The intra-domain experiment shows our method is competitive with Kea++ algorithm, and not domain-specific.

**Keywords** Keyphrase extraction, Semantic relatedness, Word sense disambiguation

## 1 引言

关键词提供了文档概要的信息,在检索系统、文本聚类 and 分类系统中被大量使用。关键词描述的好坏直接决定了这些系统的性能。在最为理想的情况下,关键词应该是人为给出的。但是,随着 Internet 的发展,人为给出文档的关键词是不现实的,所以,高性能的关键词自动提取算法的研究是十分重要的。目前大部分的关键词提取算法都是基于机器学习的方法<sup>[1,2]</sup>。在这些算法中,我们发现同一篇文章中的同一个词在不同的地方或许有着不同的意思,比如说,“mouse”能够表示老鼠或者是鼠标的意义,同样,不同的词能够表示相同的意思,比如说同义词。这些现象产生的原因在词汇层面(代表意思的词)和概念层面(意思本身)的差别,这样将会导致关键词提取的不准确。为了解决这个问题,我们使用词义<sup>2</sup>代替词来解决这个问题,因为词义只有唯一的意思。在关键字提取算法中,我们首先采用消歧算法得到关键候选词的词义,然后,在词合并、特征提取和评估的步骤中,将考虑这些词义之间的相关度来提高算法的性能。

我们的关键词提取算法同时考虑了统计和语义信息,主

要分为候选词提取和过滤两个阶段。在候选词的选取阶段,当候选词从文档中被提取出来后,消歧算法被用来得到候选词的词义,然后通过计算候选词义之间的相关度来进行词合并。在过滤阶段,我们将计算候选词义的四个特征值:TF×IDF,候选词最早出现的位置,候选词的长度以及该候选词和其他候选词间的语义相关度。然后我们将使用已知关键词的训练样本来生成一个 Bayes 的估计模型,使用这个估计模型来计算每一个候选关键词的可能性。最后,可能性最大的  $n$  个候选关键词将会被认为是最终的结果。

当评估关键词的提取算法时,现有的方法是匹配算法自动提取的关键词的词根和人为赋予的关键词的词根。这种方法很简单并且迅速,但是却不是最有效的。最主要的原因在于,这种评估的算法使用的是语法上的完全匹配而不是词义的匹配。为了克服这个缺点,我们的评估算法通过计算提取的关键词和人为赋予的关键词之间的相似度来评估关键词提取算法的性能。

这篇文章的工作是设计一种准确的关键词自动提取算法。在下一节,我们回顾相关的工作。然后详细描述基于语义的关键词提取算法。接着,证明基于语义的算法有效性的

<sup>\*</sup>)国家自然科学基金资助项目(60675015)资助。方俊 博士生,主要从事语义网和数据挖掘研究;郭雷 博士生导师,主要从事神经网络、模式识别和知识管理等;王晓东 博士生,主要从事语义网和智能检索。

<sup>1</sup> 关键词表示是很多字组成的词,而关键字表示的是单个的字。人们一般给文章提供关键词。在这篇文章中,我们调查的是关键词提取的算法。

<sup>2</sup> 在这篇文章中,我们将使用定义在 WordNet<sup>[4]</sup>中的词义,WordNet 为每一个词定义了一组词义。

评估实验将会被陈述。最后,介绍结论和未来的工作。

## 2 相关工作

因为关键词的重要性,有大量的工作在这个方面开展。最著名方法是 Kea<sup>[1,4]</sup>。Kea 采用的是朴素贝叶斯的机器学习方法来提取文档中的关键词。最初的 Kea<sup>[1]</sup> 系统使用两个特征来预测候选的关键词是不是关键词。Turney 发现候选关键词内聚性能够影响到算法的性能,所以他使用基于 Web 挖掘的候选关键词之间的统计信息来计算内聚度,并使用这个特征提高关键词提取算法的性能<sup>[2]</sup>。目前的 Kea 系统, Kea 4.0, 提取的文档集合训练的文档集在同一个领域内。通过采用基于领域词典的 Kea++ 算法,该关键词提取系统的性能有很大的提高。Kea++ 算法和我们的基于语义的关键词提取算法都使用语义的信息,但是它们有以下不同点: (1) Kea++ 是面向特定领域,它只能提取和训练文档集在同一个领域内的文档的关键词。这种领域性带来了两个缺点: 首先,当要变换领域时,必须要创建不同领域的词典;其次,因为大多数网络上的电子文档并没有分类,所以我们要手动收集训练文档。与 Kea++ 不同的是,我们的方法没有领域的限制性,不用人为创建词典和收集训练文档,所以更加灵活。(2) Kea++ 算法处理的对象是词,从上面的描述我们知道,不同词表达的意思之间的差别将会导致关键词提取的差别。我们的算法首先使用消歧算法得到候选词的词义,然后对候选词义进行后续处理,从而提高了算法的性能。(3) Kea++ 使用一种简单的基于层次的方法来估计词之间的相关度,这种方法没有考虑连接间的强度不同的问题。在我们的算法中,我们采用基于注释的相关度量方法 (gloss based measure)<sup>[5]</sup>, 这种方法通过计算不同词之间注释的相同的单词数来估计词之间的相关度。相对于 Kea++ 的方法,基于注释的方法更加成熟和有效。

## 3 基于语义的关键词提取算法

与现有的关键词提取算法不同,我们的基于语义的关键词提取算法生成的是词义的集合而不是词的集合。在这篇文章中,定义在 WordNet<sup>[3]</sup> 中的词义被用来表示词的意思。在我们的实验中,WordNet 2.1 这个最新的 windows 版本被使用,它包括了 155327 个单词和 117597 词义。

### 3.1 词义

在我们的关键词提取的方法中,词义成为了处理的对象。一般来说,在文档中,有两个类型的词: 单个词和组合词。这两种词的词义定义如下所示。

单个词仅仅包含一个能在 WordNet 中找到的词,它的词义  $S(t)$  定义如下:

$$s(t) = s_k, \text{ where } s_k \in \text{Synset}(t) \quad (1)$$

组合词本身不能在 WordNet 找到,它包含几个 WordNet 中定义的词,  $t = w_1 w_2 \dots w_n$ , 它的词义是组成它的词的词义的并集:

$$S(t) = \bigcup s_k, \text{ where } s_k \in \text{Synset}(w_k), w_k \in t \quad (2)$$

我们通过使用消歧算法来获得候选词的词义。文档中被消歧词附近的词作为上下文来判断目标词的词义。

### 3.2 选择候选关键词

在这一小节中,我们陈述候选关键词选择的标准和规则。首先,去除数字和各种标点符号,将文档中的句子分成一个个的单词;然后,使用下面一些规则来判断这些词是不是候选关键词: 候选关键词有最大的长度限定,它不能是大写的固有名

称,它的开头和结尾不能是停用词,最后,词中所有的大写字母变成小写字母,并采用迭代的 Lovins 方法来提取每一个词的词根。Lovins stemmer<sup>[7]</sup> 方法将会除去单词的前后缀,并且不断地重复这个过程,直到单词不发生变化,这样就得到了单词的词根。转换大小写和词根,能消除同一单词不同变化形式的影响。

#### 3.2.1 候选关键词消歧

经过上面的处理以后,我们将使用消歧算法得到候选关键词的词义。词义是由它周围的单词所决定的。消歧算法计算被消歧的词所有的可能词义和它周围单词的语义相关度,并且认为相关度最大的词义就是该词在现在语境下的正确的词义。这种算法的依据在于,同时出现的相近的词应该在语义上有某种程度的相关性。本文中,我们将采用 extended gloss overlap<sup>[5]</sup> 语义相关度算法来对候选关键词进行消歧的处理。在文献[6]中,extended gloss overlap 算法已被证明了是一种非常有效的语义相关度和消歧的方法。该算法通过计算词之间的注释相同的单词个数,以及在 WordNet 中和词相关联的词的解释的单词重叠数来得到它们之间语义相关度的值。

在对候选关键词消歧的过程中,大小为  $W$  的窗口内的单词将会被选择作为目标词的上下文。假设这个上下文的单词集合为  $C$ , 被消歧的目标词为  $t$ , 它的可能的词义  $S(t)$  由公式 (1) 和 (2) 所定义。

当目标词是单个词时,我们使用公式 (3) 计算所有可能词义与上下文集合之间的语义相关度的大小,该公式将可能词义和上下文集合所有元素之间的语义相关度相加,我们使用  $SenseScore_k$  来表示可能词义  $k$  的值的大小。当目标词是组合词时,采用公式 (4) 得到语义相关度。

$$SenseScore_k = \sum_{i=1}^W rel(s_k, c_i) \quad (3)$$

$$SenseScore_k = \sum_{j=1}^{|S_k|} \sum_{i=1}^W rel(s_j, c_i) \quad (4)$$

计算两个词之间语义相关度的函数 relatedness() 采用的是 extended gloss overlap 算法。最终,具有最大  $SenseScore$  值的词义是目标候选关键词的正确词义。

Extended gloss overlap 算法通过计算词义之间以及在 WordNet 中和词义相关联的词义的解释的单词重叠数来得到它们之间语义相关度的值。我们使用公式 (5) 来对其进行归一化处理。从这个公式我们可以看出,两个词之间的相关度处于 0 和 1 之间,同义词之间的语义相关度值为 1。

$$rel(s_i, s_j) = \frac{\text{number\_of\_overlaps}}{(\text{wordNumInGlossOf } s_i + \text{wordNumInGlossOf } s_j) / 2} \quad (5)$$

词和词之间以及词和词义之间的语义相关度的值是和这些词相对应的词义之间语义相关度的最大值。

#### 3.2.2 基于语义的候选词合并

这个步骤将会对候选关键词进行基于语义的合并。我们首先计算关键词之间的语义相关度,然后引入一个阈值  $\alpha$ , 如果两个词的相关度大于  $\alpha$ , 则认为这两个词在语义上是相同的,我们将它们当作同一个词来处理。

### 3.3 特征选择和离散化

我们的机器学习的方法需要定义一些特征,从训练的文档中建立模型,并用这个模型来预测新的文档中的关键词。本文中,我们选择了四个特征: TF × IDF, First occurrence, Length 和 Coherence。

TF × IDF 表示一个词在文档中出现的频率,并且和训练集中出现该词的文档数作比较。如果一个词语的 TF × IDF

<sup>3</sup> Kea 的更多信息可以参考 <http://www.nzdl.org/kea/>

值越高,则表示这个词越有可能是关键词。First occurrence 表示词语在文章中第一次出现的平均位置。出现在开头和结尾的词语是关键词的概率比较大。Length 代表的是组成词语单词的个数。以上这三个特征在 Kea++ 算法中也同样被用到了。在文献[2]中,Coherence 特征被证明能够很大程度地提高关键词提取的性能。与文献[2]使用的基于统计的方法不同,本文采用基于 WordNet 计算的语义相关度来决定词语之间的内聚性。假设集合  $S$  表示候选的关键词义,我们将词义  $s_i$  和集合中其他词义的语义相关度相加得到  $s_i$  的内聚度  $coherenceScore$  的值,如公式(6)所示:

$$coherenceScore_{s_i} = \frac{\sum_{i=1, j \neq i}^{|S|} (s_i, s_j)}{|S| - 1} \quad (6)$$

在公式(6)中,分母用来对内聚度的值进行归一化处理。如果一个候选关键词义的内聚度越大,则表示它和候选关键词义集合中的其他词义关联程度越大。

以上四个特征都是连续的量,为了机器学习的方法能够使用,我们使用 equal-depth partitioning<sup>[8]</sup> 的方法将这些连续的量转变成离散的量。该离散化的方法划分成  $N$  个区间,每一个区间包含几乎相等的样本。

### 3.4 关键词的提取

我们将使用上面的四个特征值从训练的文档中建立朴素贝叶斯模型。对于每一篇训练集中的文档,我们都会先采用上面陈述的步骤先进行处理,然后,对于每一个候选关键词,我们将会使用下面的两个公式来计算它的两个数值:

$$Pr[yes|T, O, L, C] = \frac{Pr[T|yes] \times Pr[O|yes] \times Pr[L|yes] \times Pr[C|yes] \times Pr[yes]}{Pr[T, O, L, C]} \quad (7)$$

$$Pr[no|T, O, L, C] = \frac{Pr[T|no] \times Pr[O|no] \times Pr[L|no] \times Pr[C|no] \times Pr[no]}{Pr[T, O, L, C]} \quad (8)$$

$Pr[T|yes]$ ,  $Pr[O|yes]$ ,  $Pr[L|yes]$  和  $Pr[C|yes]$  分别表示候选关键词是关键词时的 TF×IDF, First occurrence, Length 和 Coherence 的值,  $Pr[yes|T, O, L, C]$  表示该词语是关键词的概率。为了对其进行归一化处理,我们引入了分母  $Pr[T, O, L, C]$ 。公式(8)的解释与此类似,它得到该词语不是关键词的概率。最后,该词语成为关键词的概率可以用公式(9)计算得出:

$$Pr = \frac{Pr[yes|T, O, L, C]}{Pr[yes|T, O, L, C] + Pr[no|T, O, L, C]} \quad (9)$$

候选关键词通过  $Pr$  来进行排序,按照顺序,将用户所需要的数量的关键词返回给用户。

## 4 算法评估

目前对关键词提取算法的评估方法是将算法提取出来的关键词与标准的人为提取出来的关键词作词法上的匹配。因为这种方法没有考虑语义的关系,所以它是不充分的。本文中,我们提出一种基于语义的评估方法。

### 4.1 基于语义的评估方法

著名的 Precision(查准度), Recall(查全率)和平衡它们两者的  $F$ -measure 被用来对我们的关键词提取算法进行评价。

$$Precision = \frac{\text{correct extracted keyphrases}}{\text{all extracted keyphrases}} \quad (10)$$

$$Recall = \frac{\text{correct extracted keyphrases}}{\text{manually assigned keyphrases}} \quad (11)$$

$$F\text{-measure} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (12)$$

本文中,这3个度量值的计算方法与传统的方法有一些不同。对一个词语是否是正确的关键词的评判的准则是基于语义上来考虑的。我们首先采用公式(5)计算词语之间的相关度,然后,我们引入阈值  $\beta$  来判断词语是否相等。公式(10),(11)和(12)被用来对每一篇文档进行评估,最终的 Precision, Recall 和  $F$ -measure 是整个测试文档集的平均值。为了证明我们的基于语义的评估方法的有效性,我们让6个大学生对20篇文档赋予相同数量的关键词,学生们对这些文档的内容非常熟悉。我们设定某一个学生赋予的关键词为标准,然后计算其他学生赋予的关键词的 Precision, Recall 和  $F$ -measure 的值。我们采用传统的词法匹配的方法和基于语义的方法来计算。在基于语义的方法中,阈值  $\beta$  被设定为0.9。结果传统方法的平均  $F$ -measure 值为0.830,基于语义方法的平均  $F$ -measure 值为0.962。这个简单的实验可以证明基于语义的评估方法比传统的评估方法更加有效,因为不同的人给同一篇文档赋予的关键词可能在语法上不同,但在语义上应该有很大的相关度。

我们进行了两个实验来比较我们的关键词提取算法和 Kea 算法。第一个实验训练和测试集跨越了多个领域,第二个实验训练和测试集在一个领域。阈值  $\alpha$  和  $\beta$  被设定为0.9。要注意的是,这些实验的评估方法采用的是基于语义的评估方法。

### 4.2 跨领域的实验

在这个实验中,我们从网上下载了200篇被赋予了关键词的文档,这些文档平均有5.3个关键词,它们涉及到了物理、化学、计算机、经济和生物领域。随机地选取150篇文档作为训练文档,剩下的50篇文档用来进行测试。

我们对三种关键词提取方法进行比较:Kea 算法、我们的基于语义的关键词提取算法和仅仅采用前3种特征值的基于语义的算法。比较第三种提取算法的原因在于 Kea 算法也仅仅使用了这三种特征值:TF×IDF, First occurrence 和 Length,这样能更清楚地看出考虑语义信息后的好处。图1给出了实验的结果。横轴表示总共产生的关键词的个数,它的范围是从1到20,纵轴表示算法提取出来的关键词在语义上正确的个数平均值。从图中我们可以看出,考虑4个特征值的基于语义的算法有最好的性能,接下来是考虑三个特征值的基于语义的算法,性能最差的是 Kea 算法。我们从每篇文档自动提取出的关键字提取集合中按顺序选取前5位关键词,因为这是文档平均包含的关键字的值,然后,采用评估公式(10),(11)和(12)来计算每篇文档的 Precision, Recall 和  $F$ -measure,最后,我们计算它们的平均值,最终的结果如表1所示。从结果中,我们可以看出,当考虑到语义信息后,关键词提取的性能能够得到很大提高。

### 4.3 同一领域内的实验

我们采用同样的方法来对我们的算法和 Kea++ 算法进行比较,稍稍不同的是,因为平均关键词个数为5.9的缘故,我们选取前6位的关键词来进行评估,实验的结果如图2和表2所示。从实验结果可以看出我们的算法的性能和 Kea++ 算法的性能非常接近。我们的算法比 Kea++ 算法稍稍差一点的主要原因在于,Kea++ 使用的是农业领域的专用词典,而我们的算法使用的是包含多领域的通用 WordNet 词典。从跨领域的实验和同一领域的实验,我们还可以发现,

表1 基于语义的关键词提取算法和 Kea 算法性能的比较

|         | Precision | Recall | F-measure |
|---------|-----------|--------|-----------|
| Kea 算法  | 0.255     | 0.230  | 0.242     |
| 三特征语义算法 | 0.375     | 0.386  | 0.380     |
| 四特征语义算法 | 0.531     | 0.555  | 0.543     |

我们的基于语义的算法在性能上是差不多的,这表明我们的算法没有领域相关性,所以比 Kea ++ 算法有着更好的应用前景。

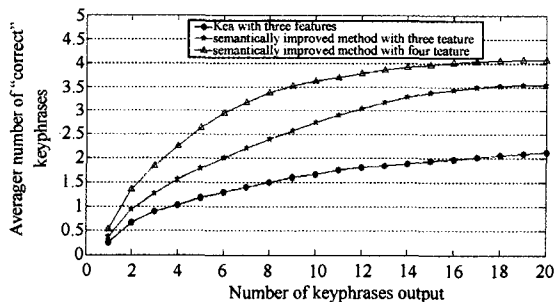


图1 基于语义的关键词提取算法和 Kea 算法的比较

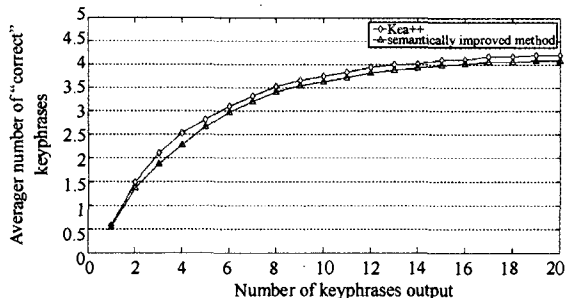


图2 农业领域中基于语义的提取算法和 Kea ++ 算法的比较

**结束语** 在这篇文章中,我们提出了一种考虑词语语义信息的关键词提取算法。该算法首先使用消歧算法得到候选关键词的词义,然后在后面的步骤中使用这些词义的语义相

关度信息。实验表明通过考虑语义的信息,关键词算法的性能能得到很大的提高。同时,相对于 Kea ++ 算法,我们的算法没有领域的限制性。在未来的工作中,我们将会采用更多的数据来对基于语义的算法进行测试。另外,现有的消歧算法的精度不是特别高<sup>[6]</sup>,因此我们计划设计一种更加有效的消歧算法来提高基于语义的关键词提取算法的性能。

表2 基于语义的关键词提取算法和 Kea ++ 算法性能的比较

|           | Precision | Recall | F-measure |
|-----------|-----------|--------|-----------|
| Kea ++ 算法 | 0.575     | 0.569  | 0.572     |
| 基于语义的算法   | 0.550     | 0.566  | 0.558     |

### 参考文献

- [1] Witten I H, Paynter G W, Frank E, et al. KEA: Practical automatic keyphrase extraction // Proc. DL '99, 1999; 254-256
- [2] Turney P D. Mining the Web for Lexical Knowledge to Improve Keyphrase Extraction: Learning from Labeled and Unlabeled Data. Technical Report ERB-1096. National Research Council Canada, 2002
- [3] Fellbaum C. Wordnet: An Electronic Lexical Database. Cambridge: MIT Press, 1998
- [4] Medelyan O, Witten I H. Thesaurus Based Automatic Keyphrase Indexing // Proc. of the Joint Conference on Digital Libraries 2006. Chapel Hill, NC, USA, 2006: 296-297
- [5] Banerjee S, Pedersen T. Extended gloss overlaps as a measure of semantic relatedness // Proceedings of the Eighteenth International Conference on Artificial Intelligence. Acapulco, 2003; 805-810
- [6] Pedersen T, Banerjee S, Patwardhan S. Maximizing Semantic Relatedness to Perform Word Sense Disambiguation. Supercomputing institute research report umsi 2005/25, University of Minnesota, 2005
- [7] Lovins J B. Development of a stemming algorithm. Mechanical Translation and Computational Linguistics, 1968, 11: 22-31
- [8] Dougherty J, Kohavi R, Sahami M. Supervised and unsupervised discretization of continuous features // Proceeding of ICML-95, 12th International Conference on Machine Learning. Lake Tahoe, US, 1995; 194-202

(上接第 147 页)

trieval Test Collection (Version 2.1)<sup>[15]</sup>, 以 communication, economy, education, food, medical, travel, weapon 这 7 个域本体所包含的概念作为参数来源, 服务样本集数量分别为 135, 52, 25, 106, 29, 206, 25, 在这 5 个服务样本集上所进行的服务请求数目分别为 6, 1, 1, 2, 6, 11, 1。实验结果如图 2 和图 3 所示。

从图 2 和图 3 可以看出, 无论是查准率还是查全率, 基于语义的方法都远高于基于关键字的匹配方法, 三种方法的平均查全率分别为 27.3%, 75.8%, 93%; 三种方法的平均查准率分别为 19%, 68.7%, 89.3%。显然, 本文所提出的基于服务分层的匹配方法在查全率和查准率上都高于传统的基于语义的网格服务匹配方法。

**实验三: 基于服务分层的匹配方法和直接基于 OWL 推理机方法的时间开销比较**

本实验比较基于服务分层的方法和直接基于 OWL 推理机方法的时间开销, 仍以文献<sup>[14]</sup>中的方法为代表, 实验设置与实验一相同。实验结果如图 4 所示。

从图 4 可看出: 基于服务分层的匹配方法和直接基于 OWL 推理机的服务匹配方法相比, 虽然在服务发布阶段因需要构造概念分层和服务分层而增加了一定的时间开销(约为 6 倍), 但用户请求响应时间却大大缩短了, 平均响应时间为 8.9ms。因为服务发布阶段对实时性要求不高, 而即时的服务请求对实时性要求很高, 所以基于服务分层的匹配方法更能满足实时服务匹配的需求。

**结束语** 针对传统的基于关键字的网格服务匹配方法所存在的灵活性差、查全率和查准率低等不足, 本文提出了一种新的基于本体的网格服务匹配方法, 该方法利用本体来描述网格服务的语义信息, 同时, 利用 OWL 推理机对网格服务进

行服务分层, 以提高服务匹配的效率。实验结果表明, 本文所提出的网格服务匹配方法与传统的基于关键字匹配的服务匹配方法相比, 具有较高的查全率和查准率, 同时, 与直接基于 OWL 推理机的语义网格服务匹配方法相比, 更能满足实时服务匹配的要求。

### 参考文献

- [1] Foster I, Kesselman C, Nick J, et al. The physiology of the grid: An open grid services architecture for distributed systems integration. <http://www.globus.org/research/papers/ogsa.pdf>, 2002
- [2] UDDI: The UDDI Technical White Paper. <http://www.uddi.org>, 2000
- [3] Globus project. <http://www.globus.org>
- [4] Lee T B, Hendler J, Lassila O. The Semantic Web 1 New York: Scientific American, 2001
- [5] 李善平, 尹奇, 胡玉杰, 等. 本体论研究综述. 计算机研究与发展, 2004, 41 (7): 1041-1052
- [6] Studer R, Benjamins V R, Fensel D. Knowledge Engineering, principles and methods. Data and Knowledge Engineering, 1998, 25 (12): 161-197
- [7] Song Zilin, Ai Weihua, Wang Yi, et al. Service Search Strategy Based on Graph in Grid Environment // Proceedings of the Second International Conference on Semantics, Knowledge, and Grid (SKG'06)
- [8] Zhang Y, Song W. Semantic Description and Matching of Grid Services Capabilities
- [9] Ludwig S A, Reyhani S M S. Semantic Approach to Service Discovery in a Grid Environment. Journal of Web Semantics, 2006
- [10] 史忠植, 蒋运承, 等. 基于描述逻辑的主体服务匹配. 计算机学报, 2004, 5(17)
- [11] Fact++. <http://owl.man.ac.uk/factplusplus/>
- [12] Paolucci M, Kawamura T, Payne T R, et al. Semantic Matching of Web Service Capabilities. Lecture Notes in Computer Science, 2002, 2342: 333-347
- [13] Ludwig S A, Reyhani S M S. Introduction of semantic matchmaking to Grid computing. Journal of Parallel and Distributed Computing, 2005, 65: 1533-1541
- [14] <http://www.dfki.de/scallops>