基于 Wikipedia 的语义元数据生成*

韩先培 赵军

中国科学院自动化所模式识别国家重点实验室 100190

E-mail: xphan@nlpr.ia.ac.cn

摘　要：语义元数据提供数据的语义信息，在数据的理解、管理、发现和交换中起着极为重要的作用。随着互联网上数据爆炸式的增长，对自动元数据生成技术的需求也就变得更加迫切。获得目标语义元数据及得到足够的训练语料是使用自动生成技术的两个基本问题。由于获得目标语义元数据需要专家知识，而获得足够的训练语料需要大量的手工工作，这也就使得这两个问题在构建一个成功的系统时至关重要。本文基于 Wikipedia 来解决这两个问题：通过分析一个类别中条目的目录表(table-of-contents)来抽取目标语义元数据，通过对分析文档结构和赋予目标结构正确的语义元数据来构建训练语料库。实验结果表明，本文的方法能够有效的解决这两个问题，为进一步的大规模的语义元数据应用系统打下了坚实的基础。

关键词：元数据，语义元数据，数据处理，语料库构建，语义标注

Semantic Metadata Generation: A Method Based on Wikipedia

HAN Xianpei, ZHAO Jun

National Laboratory of Pattern Recognition Institute of Automation, Beijing 100190

E-mail: xphan@nlpr.ia.ac.cn

Abstract: Semantic metadata, which provides semantic information about data, plays an important role in document management, fusion and information search. The automatic metadata generation technique has been a big challenge in the data explosion time. There are two fundamental problems in using automatic techniques: the acquisition of target semantic metadata and the obtaining of training corpus. The first problem involves expert knowledge and the second problem needs lots of manual work, these make these two problems critical in building a successful system. In this paper, we resolve the two problems based on Wikipedia: we extract the target metadata by analyzing the table-of-contents of Wikipedia's entries and build the training corpus by analyzing the Wikipedia entry's structure and assigning the true semantic metadata. The experiment results demonstrated that both problems can be resolved effectively.

Keywords: metadata, data processing, semantic metadata, Corpus building, semantic annotation.

1　引言

海量的信息以非结构化文本的方式存在([1])，其中大部分信息仅能为人所阅读及理解。与此同时，信息社会需要大量机器可直接处理的数据和信息。因此，如何将非结构化的数据转换成机器可处理的结构化数据也就成为了一个重要的课题。语义元数据提供数据的语义信息，将仅仅为人所能阅读信息转换为机器可处理的信息中，起着极为重要的作用。这也使得近年来给数据赋予其语义信息的工作，既语义元数据生成，又叫语义标注，得到了越来越多研究人员的关注。

不幸的是，手工创建语义元数据需要花费大量的时间和精力 ([4])，同时考虑到互联网上信

*本文受国家自然科学基金项目（60673042）、国家 863 计划项目（2006AA01Z144）和北京市自然科学基金项目（4073043）资助

息的量及其增长速度，通过手工来完成这个工作也就变得不可能。因此，在语义元数据的创建过程中，需要研究自动语义标注的技术。机器学习领域的发展，给计算机提供了强有力的工具来自动完成语义元数据生成工作。通常，在使用机器学习技术来构建自动元数据生成系统时，需要关注以下的几个问题：

1.确定需要抽取的目标语义元数据的集合；

2.构建一个模型来模拟元数据的生成过程；

3.获取足够多的语料，训练上面的模型，使得这个模型产生的结果尽可能的正确。

机器学习技术为我们提供了解决第二个问题的模型。但是，在传统的解决方法中，确定抽取的目标语义元数据需要专家的参与，获取训练语料需要大量的人工，这也就使得这两个问题在构建一个自动语义元数据生成系统时至关重要。随着 Web2.0 的发展，大量用户协作式生成的高质量的数据被公开提供，其中的一个杰出代表就是 Wikipedia[1]，这给我们提供了解决这两个问题的一种新的方法：从这些数据中抽取我们需要的知识及语料。

本文展示了如何使用Wikipedia提供的数据来解决自动语义元数据生成中的第一个和第三个问题，并展示了在这个过程中需要的问题和相关的技术。我们通过如下的两个阶段来依次解决第一个和第三个问题：

1. 第一阶段：通过分析 Wikipedia 中一个类别的条目的目录表(table-of-contents)来得到该类别的目标语义元数据的集合，同时使用了一个结构推理算法来分析这些语义元数据的结构关系；
2. 第二阶段：通过分析 Wikipedia 的每一个条目的结构并且将元数据赋予页面的子结构来构建训练语料。

本文的剩余部分安排如下：在第二节中，我们简要的回顾了语义元数据生成的相关工作；在第三节中，我们详细的描述了我们的工作及方法；在第四节中，我们展示了使用我们的方法获得的“国家”领域的训练语料库及目标语义元数据及相关的一系列性能实验结果；在第五节中，我们对我们的工作进行综述并对将来的工作进行了展望。

2 相关工作

在本小节中，我们对自动元数据生成的各种相关工作进行一个简要的回顾并分析其优缺点，我们还简要介绍了 wikipedia。

2.1 自动元数据生成相关工作

手工语义元数据生成耗费大量的时间且包含大量错误[4]，这使得一些工作集中在研发标注工具来简化手工语义元数据生成的工作（[3][4][5][6][7][8][16]）。虽然这些工具能够简化语义元数据的手工标注的工作，但是还是需要依靠人来完成工作。

为了降低语义元数据生成过程中人的参与，一些半自动的辅助技术被应用上述标注工具中。CREAM 系统的扩展，S-CREAM([9])，通过从手工标注的实例中抽取规则来自动处理跟以前标注实例相近的情况，同时也学习了一个包装器（wrapper）来处理那些结构近似的网页。在[17]中提出的 PANKOW 方法，CREAM 的模块之一，使用一个无监督基于模式的方法来将实例匹配到一个给定的 Ontology 当中，但是，使用的模式相当简单且受到限制。上述这些技术，虽然能够一定程度上降低对手工的依赖，但都仅仅能够起到辅助的作用。

为了彻底的自动化整个标注过程，大量的工作集中在设计自动语义元数据生成模型与方法上面。Chien-Chung Huang 等人([2])提出了一个基于 Web 的自动主题元数据生成系统：首先

[1] http://www.wikipedia.org

通过 Hier-Concept Query Formation 方法来构建每一个种类的主题的向量表示，接着一个文本片段被送到搜索引擎并利用返回结果来构建该文本的向量表示，最终最近邻方法被用来确定文本片段的主题。Hsin-Chang Yang 和 Chung-Hong Lee 在[11]中提出了一种自动生成网页的语义元数据的方法：网页被自组织聚类算法聚成不同的类别，文本挖掘技术被应用到每一个类别中来抽取每一个类别的语义描述。A. Dingl 等人在[12]中提出了一个叫做 Armadillo 的框架：首先从结构化的数据源中抽取一小部分种子信息，然后通过一个 bootstrap 的过程来标注领域特定的语义信息。H. Graubitz 等人[14]展示了 DIAsDEM 框架，这个框架能够自动的标注文档的语义信息并且生成给定领域的 XML 的 DTD 文件。这个工作在[10]中得到了扩展：DTD 从一个平坦的结构转换为了层次的结构。J. Li [15] 展示了一个基于依存语法的将句子进行语义标注并转化为 RDF 样式的机器学习方法。Dill 等人[13]描述了 SemTag，一个在大规模语料里面进行语义标注的工程，主要集中在检测网页实体的共现上面。上面的这些工作，尽管以实现了一定层次的自动化，但实际中仍然需要依靠人的参与：要么需要手工标注大量的语料，要么需要人工参与到整个过程当中的一部分，尽管如此，这些工作仍然大大的自动化了语义元数据生成的过程。

2.2 Wikipedia

Wikipedia 是一个使用 wiki 软件来协作式创建一个大百科全书的项目。使用基于快速编辑的 wiki 语法，所有人都能方便的参与这个项目。这使得 Wikipedia 得到了极大的发展，到 2006 年 10 月，Wikipedia 的英文版的条目已经超过了 140000000 条，拥有近五万人的庞大活跃用户[2]。Wikipedia 已经成为互联网上最大的数据源之一。Wikipedia 使用条目的来组织所有内容，每一个条目都有其单独的 URL 地址。除了一些辅助性条目之外，每一个条目都对应着一个概念或者一个实体，如“Republic of China”。每一个条目具有特定的种类。条目之间通过内部的链接来相互连接。

3 基于 Wikipedia 的语义元数据生成

在本小节中，我们介绍利用 Wikipedia 来解决语义元数据生成中的目标语义元数据及训练语料问题的相关工作。其总体工作框架如图 1 所示。工作被分成两个部分：（1）目标语义元数据抽取；（2）语料库构建。本文通过对抽取一个类别的条目的目录表中的语义元数据来解决目标语义元数据的问题，通过抽取条目的文档结构并赋予正确的语义元数据来构建训练语料库。下面我们分别详细介绍这两个部分工作。

3.1 目标语义元数据抽取

目标语义元数据抽取的目标是结构化的一个目标语义元数据集，对象是 Wikipedia 网页中的目录表（table-of-contents），图 2 中是 Wikipedia 中两个目录表的例子。通常目录表中包含了丰富的语义元数据信息，如图 2 中的“History”以及“Politics”，而且同一个种类的目录表通常包含许多相同的语义元数据，在图 2 中 “Geography”及“History”同时出现在两个目录表中。而且，不同的元数据的关系也被包含在小节标题与子小节标题的关系中。因此，目录表是一个良好的语义元数据源。下面，我们介绍从目录表中抽取结构化语义元数据的两个步骤：

[2] http://stats.wikimedia.org/EN/Sitemap.htm

Step 1: 从小节标题中抽取出语义元数据；

Step 2: 使用结构化推理算法，将所有的元数据组织成层级(hierarchical)结构。

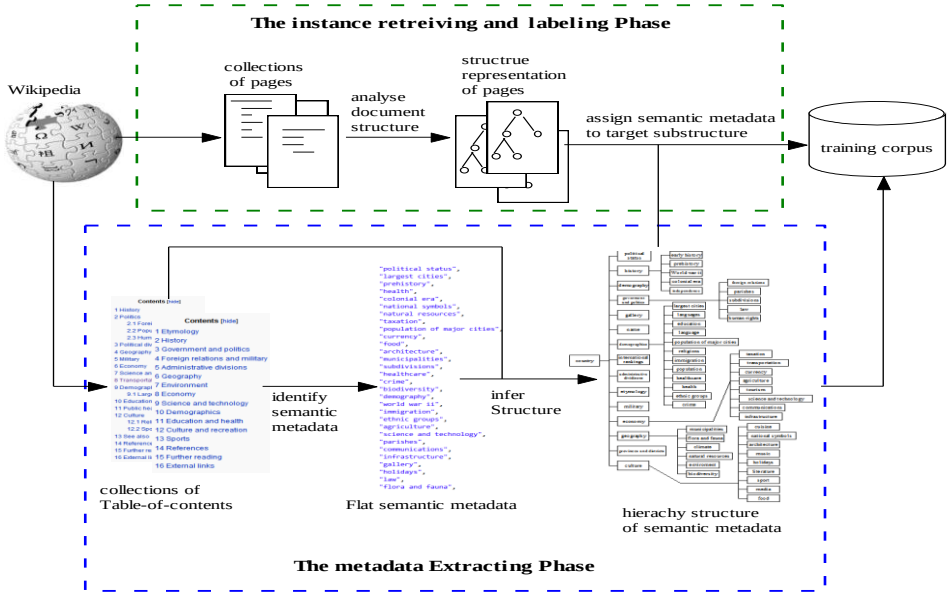

图 1. 工作框架图

Fig.1 The working framework

图 2. “China”页面及“Japan”页面的目录表

Fig.2 The Table-of-Contents of “China” and “Japan” entry

3.1.1 抽取语义元数据

尽管从目录表的小节标题中抽取语义元数据有上面所说的各种优点，但是相比于语义元数据，这些小节标题还存在如下的几个缺点：

1） 不够统一。如“administrative divisions”与“administrative division”应该表示的是同一

个语义元数据。

2） 存在着并列结构。如“Foreign relations and military”实际上是两个语义元数据：“Foreign relations”和“military”。

3） 有的小节标题对于作为一个类别的语义元数据来说过于特殊。如“World War I and the Armenian Genocide”就对于国家条目来说过于特殊。

针对以上的几个问题，我们使用如下的算法从小节标题中抽取语义元数据：

1） 归一化。我们使用词干分析器将所有的小节标题都调整为同一形式。

2） 并列结构分析。由于小节标题中的并列结构较为简单，我们使用自然语言处理软件包 OpenNLP tool set[3]来分析小节标题中的连词结构。

3） 基于频率的过滤。

此时剩下的语义元数据，将作为最终的目标语义元数据返回。

3.1.2 将语义元数据组织成一个层级结构

在给定一个领域的语义元数据之后，下一步的工作是将这些语义元数据组织成一个层级结构。这一步通常是有益且必须的，因为不同层次的语义元数据通常表示了信息在不同粒度的语义，适合在不同层次的需求。同时语义元数据的结构也有助于通过语义元数据之间的关系来提高系统的性能。为了将抽取出的语义元数据组织成一个层级结构，我们做如下的假设：认为这个层级结构仅仅是一个两层的结构。这是个合理的假设因为 Wikipedia 的几乎所有目录表都是不超过两层的。基于这个假设，我们的结构推理算法就可以简化成如下两步：

1） 将语义元数据按照其出现的统计性质将其分类到第一层和第二层中去；

分类基于如下的两个观察：a) 第二层的语义元数据通常比第一层的语义元数据出现的次数更少；b) 第二层的语义元数据通常作为第二层的小节标题出现。我们使用公式（1）来对一个语义元数据作为第一层的语义元数据的可能性进行打分，其中 $C(s)$ 为语义元数据 s 出现的次数，$C_1(s)$ 为 s 作为第一层小节标题出现的次数，α和β为权重：

$$Score(s) = \alpha C(s) + \beta C_1(s) / C(s) \qquad (1)$$

当这个公式的打分超过一个阈值时，我们认为这个语义元数据是一个第一层的语义元数据，低于这个阈值将被分类到第二层。

2） 对每一个第二层的语义元数据，分析其上位的语义元数据。

[3] http://opennlp.sourceforge.net/

在确定语义元数据的层次之后，第二步需要确定第二层的语义元数据的上位语义元数据。对每一个第二层的语义元数据，我们使用如下的公式 2 来对一个第一层的语义元数据是其上位元数据的可能性进行打分，其中 Score(s,u)表示语义元素 u 作为语义元数据 s 的上位元数据的打分，U(s,u)是 u 作为 s 的上位语义元数据的次数，C(s)是 s 出现的次数，U(u)是 u 作为其他的语义元数据的上位元数据的次数，α和β为权重：

$$Score(s,u) = \alpha U(s,u)/C(s) + \beta U(s,u)/U(u) \quad (2)$$

我们选取打分最高的语义元数据 u 作为 s 的上位语义元数据。

至此，我们已经从一个种类的 Wikipedia 条目的目录表中，得到了语义元数据抽取系统所需要的结构化的目标语义元数据。

3.2 语料库构建

要构建一个训练语料库，需要得到相关实例并赋予实例正确的语义元数据。通常，语料中的一个实例对由（实例， 标记）对组成。在语义元数据生成系统中，由于生成通常在文章的某一个级别上进行，实例就是该级别对应的文章子结构，通常为句子或者段落。因此在构建语料的过程中：第一步需要得到实例，第二步需要给实例赋予正确的语义元数据标记。在本文的工作中，我们通过分析 Wikipedia 条目的文档结构来得到实例，然后根据文档结构所在的小节的标题来给赋予实例正确的语义标记，从而最终构建需要的训练语料库。

3.2.1 Wikipedia 条目的文档结构分析

语义元数据生成通常在文档的某一个层次的结构上进行，这就需要能够快速的得到文章的各个层次的结构。但是跟其他基于 Wiki 的信息系统一样，Wikipedia 也使用了 Wiki 标记语言，这就使得文档的结构通常不是那么明显。因此，有必要通过一个额外的分析步骤来得到 Wikipedia 的条目的结构。

已有许多工作关注如何将 Wiki 标记语言的页面转换成一个按文档的层次组织的 XML 文件。在本文中，我们使用了 Java Wikipedia API[4]来将 Wikipedia 的网页转换成一个 DocBook[5] 样式的 XML 并滤除了其中的不需要的文档结构，仅仅保留了如下的两个结构：段落和小节。同时，我们使用了 OpenNLP 的工具包对每一个段落进行了句子切分，从而最终，得到了一个小节-段落-句子的三层文档结构。

3.2.2 标注 Wikipedia 页面的文档子结构（段落和句子）

在分析了页面的文档结构之后，我们可以根据我们的语义元数据抽取的层次来得到我们的实例，下一步的工作是赋予这些实例正确的语义元数据标记。由于我们的所有语义元

[4] http://www.matheclipse.org/en/Java_Wikipedia_API
[5] http://www.docbook.org

数据都是从小节标题中抽取的来，所有，我们可以根据一个子结构所处的小节的标题来赋予它正确的语义元数据。我们使用如下的几条规则来完成标注工作：

1） 如果子结构不处于任何小节之内，则认为这个子结构的语义元数据未知，从语料中去除；

2） 如果子结构上面仅有一层的小节，则根据语义元数据抽取的几个步骤来得到这个小节标题的语义元数据，如果抽取出的语义元数据是第二层的语义元数据，则这个子结构同时也被赋予其上层的语义元数据，如果这个标题不能抽取出任何的语义元数据，则认为该子结构语义元数据未知，从语料中去除；

3） 如果子结构上面有两层的小节，则根据其第二层的小节标题抽取出的语义元数据作为其语义元数据，如果第二层的小节标题不能抽取出任何的语义元数据，则使用再上一层的小节标题来抽取其目标语义元数据。

在给页面的文档子结构标记其语义元数据之后，我们可以方便的根据语义元数据抽取所在的层次抽取出所有的（子结构，语义元数据）对来构建我们需要的训练语料库。

4 实验及讨论

在本小节中，我们实验了本文提到的方法，抽取了目标语义元数据并构建了训练语料库。同时，我们展示了元数据抽取的结果并且验证了构建的语料库的性能。实验表明，本文提出的方法能够有效的应用到自动语义元数据系统中。

4.1 数据准备

我们抽取了英文版 Wikipedia 中所有的国家条目作为我们实验的对象。这些国家条目被列在 Wikipedia 的“list of countries”条目[6]中，通过 Wikipedia 的导出 API[7]，总共下载了 221 个国家的条目数据。

同时，为了测试语料库训练的系统在其他数据集上的通用性，我们还手工标注了一个开放测试集。开放测试集的语料来自于 Infoplease 的百科全书[8]，其中包含了来自第六版的哥伦比亚百科全书中的超过 57000 篇的文章。语义元数据生成在两个级别的文档结构上（段落级和句子级）进行了实验。表 1 中是语料库和开放测试集的统计数据。

表 1. 语料库及开发测试集实例数目

Tab.1 The instance number of corpus and test set

文档结构级别	语料库样本数目	开放测试样本数目
段落级	6789	170
句子级	30670	894

[6] http://en.wikipedia.org/wiki/List_of_countries
[7] http://en.wikipedia.org/wiki/Special:Export
[8] http://www.infoplease.com/encyclopedia/

4.2 目标语义元数据识别及结构化的结果

应用 3.1 节中所述的目标语义元数据识别及结构推导算法到这 221 条“country”类别条目的 table-of-contents 上，最终得到了 60 个目标语义元数据，其结构如图 3 所示，其中 15 个语义元数据被认为是第一层的语义元数据。

从这些结果中，我们可以看出所有的语义元数据都与国家条目相关且不会过于特殊，这表明基于频率的过滤能够有效的抽取目标的语义元数据，另一方面，图 3 的所有上下位结构都比较合理，这表明通过我们的结构推理算法能够有效的推断出语义元数据之间的上下位关系。

4.3 语料库性能测试

为了测试自动生成的语料库的性能，我们设计如图 4 中的一个基于分类器的自动语义元数据生成系统：给定一个文档的子结构，一个使用 Wikipedia 中获取的语料库训练得到的分类器将每一个子结构分类到预定好的语义元数据种类中去。在本文中，我们仅仅对第一层的语义元数据进行了实验。分类器仅仅使用文档子结构当中的词（停用词被滤除）来作为特征。为了测试语料库针对不同的模型的效果，我们实验了两个不同的分类器，分别是 Naïve Bayes 和 Max Entropy。

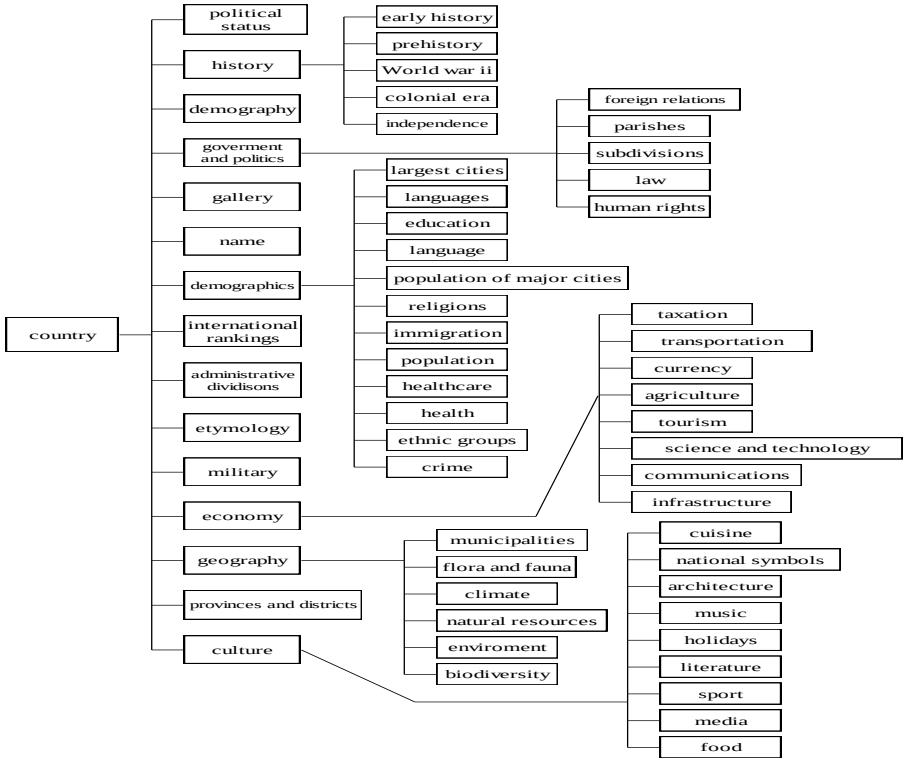

图 3. 语义元数据抽取结果

Fig.3 The result of semantic metadata extraction

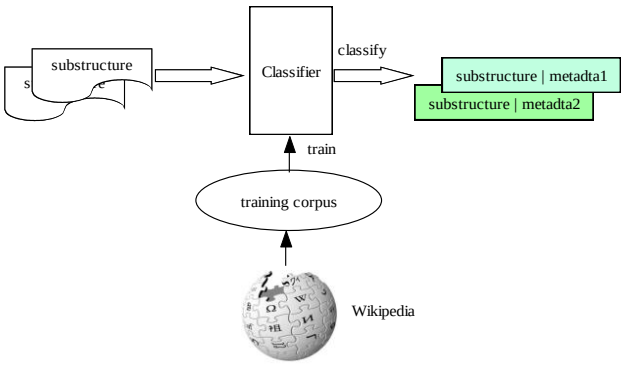

图 4.基于分类器的语义元数据生成系统

Fig.4 The classifier-based semantic metadata generation system

语料库使用 80/20 的比例来进行分割，其中 80%被用来训练，20%被用来测试。在段落上的实验结果见表 2，在句子上的实验结果见表 3。

表 2. 段落级别语义元数据生成结果

Tab.2 The semantic metadata generation result on paragraph level

分类器类别	训练集准确率	测试集准确率	开发集准确率
Naïve Bayes	0.888	0.800	0.906
Maxent	0.873	0.805	0.894

表 3. 句子级别语义元数据生成结果

Tab.2 The semantic metadata generation result on sentence level

分类器类别	训练集准确率	测试集准确率	开发集准确率
Naïve Bayes	0.827	0.731	0.779
Maxent	0.799	0.720	0.700

从以上的结果可以看出：

1. 这从语义元数据抽取的结果的高准确率上可以看出，使用本文的方法构建的语料库是内在一致的，具有良好的内部一致性；
2. 语料库具有良好的通用性，这从开放测试集上的高准确率上可以看出；
3. 语料库可以被用来构建高性能的语义元数据生成系统；
4. 相比于句子，高层的语义元数据更适合在大粒度的段落上生成，这从段落级别的准确率都高于句子级别的准确率上可以看出。这跟我们的直观想法也是一致的，通常人们在一个段落中表示一个大的主题而不是一个句子。

5 结论

在本文中，我们展示了如何使用Wikipedia来解决构建自动语义元数据生成系统的两个问题：目标语义元数据抽取及语料库的构建。实验结果表明，从 Wikipedia 出发，经过我们的语义元数据抽取方法及自动语料库构建方法，这两个问题能够自动的方便有效的解决，避免了大量的手工工作。这使得构建自动语义元数据系统变得简单和容易，从而为下一步构建大规模的语义元数据的应用系统打下坚实及良好的基础。

參 考 文 獻

[1] A.-H. Tan. Text mining: The state of the art and the challenges. *PAKDD 1999*, pages 65–70, Beijing, China.

[2] Chien-Chung Huang et al. Using a web-based categorization approach to generate thematic metadata from texts. ACM Transactions on Asian Language Information Processing Vol.3, Issue 3. 2004. pp. 190-212.

[3] S. Bechhofer and C. Gobel. Towards annotation using daml+oil. In Proc. K-CAP 2001, Canada.

[4] M. Erdmann et al.From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. COLING 2000 Workshop on Semantic Annotation and Intelligent Content, 2000.

[5] S. Handschuh and S. Stabb. Authoring and annotation of web pages in cream. WWW 2002.

[6] M.-R. Koivunen and R. Swick. Metadata based annotation infrastructure offer flexibility and extensibility for collaborative applications and beyond. In Proc. K-CAP 2001, Victoria, B. C., Canada, 2001.

[7] P. Martin and P. Eklund. Embedding knowledge in web documents. *Computer Networks*, 31:1403–1419, 1999.

[8] M. Vargas-Vera, et al. Knowledge extraction by using an ontology based annotation tool. K-CAP 2001.

[9] S. Handschuh, S. Stabb, and F. Ciravegna. S-cream–semiautomatic creation of metadata. EKAW 2002, pages 358–372, Siguenza, Spain, 2002.

[10] K.Winkler and M. Spiliopoulou. Extraction of semantic xml dtds from texts using data mining techniques. K-CAP 2001 Workshop on Knowledge Markup and Semantic Annotation, Victoria, B. C., Canada, 2001.

[11] Hsin-Chang Yang, Chung-Hong Lee, "Automatic Metadata Generation for Web Pages Using a Text Mining Approach," wiri, pp. 186-194, International Workshop on Challenges in Web Information Retrieval and Integration, 2005

[12] A. Dingli, F. Ciravegna, and Y. Wilks. Automatic semantic annotation using unsupervised information extraction and integration. In Proceedings of the 2nd International Conference on Knowledge Capture (K-CAP 2003), Florida, USA, 2003.

[13] S. Dill et al. A case for automated largescale semantic annotation. Web Semantics: Science, Services and Agents on the World Wide Web, 1(1):115–132, 2003.

[14] H. Graubitz, K. Winkler, and M. Spiliopoulou. Semantic tagging of domain-specific text documents with diasdem. In Proceedings of the 1st International Workshop on Databases, Documents, and Information Fusion (DBFusion 2001), pages 61–72, Gommern, Germany, 2001.

[15] J. Li, L. Zhang, and Y. Yu. Learning to generate semantic annotation for domain specific sentences. K-CAP 2001 Workshop on Knowledge Markup and Semantic Annotation, Victoria, B. C., Canada, 2001.

[16] S. Handschuh and S. Staab. Cream: Creating metadata for the semantic web. Computer Networks, 42(5):579–598, 2003.

[17] P. Cimiano, S. Handschuh, and S. Staab. Towards the self annotating web. In Proceedings of The 13th International Conference on World Wide Web, pages 462–471, New York, NY, USA, 2004.