中文知识本体的建立以及其在知识搜索中的应用

软件学院 杨威 5070379048
指导老师 软件学院 赵建军

摘要

传统搜索引擎往往将用户的查询假定为一种简单的模式：通过一两个关键字来告知搜索引擎用户对信息的需求，然后返回数百万的候选网页，再通过搜索排名的技术找出用户可能最需要的几个网页放在页面顶端呈现给用户。尽管目前很多关键字搜索频率都很高，但不同用户所需要的信息也许是处于不同需求的。随着个性化需求的增加，用户需要搜索引擎能够返回包含他们专业需求，个性爱好，个人健康状况等的信息。搜索这样信息的查询不仅比传统的查询多出许多的关键字，而且用户对需要返回能准确回答他们问题的信息或事实，并不是一些可能包含问题答案的网页。传统搜索引擎对这样的查询网网搜索不到匹配的网页，或者只有少许网页。而且就算其能找到所有符合信息的网页，如果所需要的信息分布在不同的网页中，则用户必须给出多个查询来收集相关的页面，然后自己提取这些页面中的相关信息并组织成一个整体。由此可见，开发一个能够解析专业化查询并返回整合信息的搜索引擎是十分必要的。

对此我们开发了中文知识本体以及知识搜索引擎。我们将网络信息的概念进行分类，对每个领域建立本体模型，并在本体模型上建立简单的知识库。我们的知识库可以回答用户能在 Wikipedia 上能找到答案的问题，对于 Wikipedia 不能解决的问题，我们会将用户的输入与我们建立的概念分类体系进行匹配，然后定向到网络其它数据源进行查询。我们目前仅对网络上结构化和部分半结构化信息进行了整合，因为我们认为语义网运动将推动 WWW 用更易被机器处理的表示方法来描述网上内容，所以我们暂时不考虑对自然语言文本的处理。

关键词：语义网，本体，知识库，资源描述框架，网络本体语言，

ABSTRACT

Queries posed to traditional Web search engines are very simple: one or two keywords to express the user's information need, and millions of matching results including many excellent hits, so that well-known ranking techniques can easily achieve high precision for the top-10 Web pages seen by the user. While this may indeed be true for the large mass of popular queries, each asked by many thousands of users, the picture is different for the long tail of individual queries about professional needs, rare hobbies, local music concerts, or personal health issues. Not only do these queries contain more keywords and return fewer results, but the user would often expect a concise answer with relevant facts rather than merely being pointed to potentially interesting Web pages. If we need information that is spread over various documents, we must initiate several queries to collect the relevant documents, and then we must manually extract the partial information and put it together. Hence, the development of a search engine which can handle such queries and integrate web information is crucial.

We describe a knowledge search engine, which depended on ontologies and knowledge base we built based on semi-structure information on Wikipedia. We also manually tag the data sources with the same taxonomy concepts. For those queries which beyond the

knowledge of Wikipedia, we match the user's keywords with a huge private concept taxonomy. Then, it routes the query to all data sources. Currently, we do not concern the information of natural language form, because Semantic Web revolution promote the WWW community to represent Web content in a form that is more easily machine-processable.
Keyword：Semantic Web，Ontology，Knowledge Base ，RDF，OWL，

1. 介绍

1.1 背景

本体这个词来源于哲学。在哲学中，本体论研究存在的本质（希腊文 $Ov\tau o\lambda o\gamma i\alpha$），即什么是存在的，已经如何描述它们。最近，计算机科学借用了这个术语，在计算机领域中，它包含了与原有含义不大相同的技术性含义。我们沿用 T.R. Gruber 提出，后经 R. Studer 改进的定义：一个本体是一个概念体系的显示的形式化规范。

在现代信息技术中，有很多应用包含了本体的背景。如机器翻译和对消除单词歧义的研究探讨了词汇知识(lexical knowledge)， 即用什么词汇表示什么概念的知识；查询语句扩展用了分类体系（taxonomy）；监督学习和半监督学习利用了本体，并且发现对背景知识的利用有利于知识问答（question answering）和信息检索（information retrieval）。当然还有语义网中所有的应用，他们几乎都包含了本体这一概念。

在我们的项目中，我们主要利用本体提高搜索引擎的搜索质量。搜索引擎可以通过本体中的概念查找相关的页面，而不是收集所有出现某些关键词的页面。

1.2 所做贡献及本文大纲

我们的目标是建立一个具有丰富知识内容的中文自然语言本体。目前现有的本体大都呈现两种趋势：一种是手工编写的本体，精度高，但覆盖率较低，另一种是自动生成的本体，覆盖率高，但精确度较低。大多数针对自然语言本体的工作都可以分为两类：一类是构建一般用途(general purpose)自然语言本体，另一类是使构建自然语言本体的工作自动化或半自动化。我们的工作侧重于前者。与大多数工作所不同的是，多数只基于一个数据源来建立他们的背景知识，如 WordNet。而我们的本体可以从多个数据源创建，我们不仅有诸如 WordNet 中文字的概念，而且还有具体的实体，如人，组织机构，地理位置，书，歌曲，产品等。更重要的是，我们突出了实体间的关系，我们存储了诸如“坐落在哪儿”，“出身在何时”，“获得何种具体奖项”这类具体的关系。实际上，我们的本体是基于自然语言，但涵盖广泛的，基本可以覆盖大多数基于一般用途的查询。同时，我们希望能通过这样的本体，来提高语义网应用的效率。为证明这样的本体是重要且有效的，我们开发了知识搜索引擎，通过借助本体及知识库来完成网络搜索。

本文之后的部分结构如下：我们将在第二部分阐述本体在语义网环境下的网络查询中所起的重要意义，并且，我们会介绍我们所做的自然语言本体，在这个查询中的作用。在第三部分，我们会介绍本体创建的方法。在第四部分，我们将阐明我们对本体语言的选择。在第五部分，我们将介绍规则逻辑的建立并描述我们所创建的本体。在第六部分，我们将介绍我们的本体是如何处理有时效性的知识。在第七部分，我们展示了一下我们的成果。在第八部分，我们会简述这方面存在的问题，以及中文本体创建的困难，并且我们将提到我们接下来要做的工作。

2. 本体与网络查询

我们主要探讨进行网络查询时需要用到的本体与相关技术，其中涉及到的本体主要有自然语言本体（natural language ontology），领域本体（domain ontology）还有本体实例（ontology instance）。

基于语义的网络查询的基本流程可以从图 1 中给出。一开始，先是由用户输入自然语言的查询语句，然后会在自然语言本体和领域本体的帮助下，被转化为规范的标准查询语句。网页是属于某些领域本体的实例，并且含有被这些领域本体进行语义标识的数据。那么，我们的目标就是要创建查询语句的概念，领域模型，及网络上本体实例这三者间的连接关系。

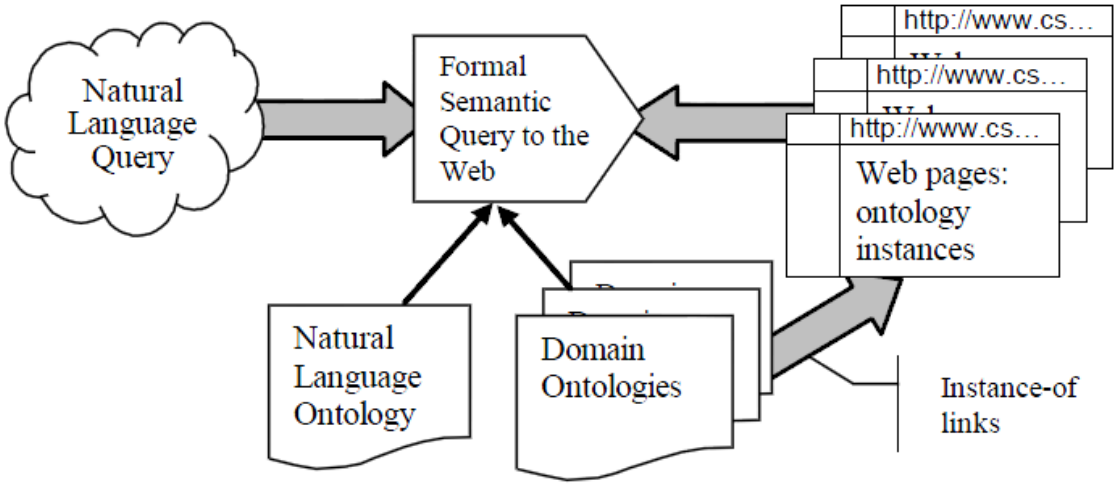


图 1　基于语义的网络查询

我们的工作主要集中在自然语言本体（natural language ontology，NLO）上。自然语言本体通常是一些表现系统背景知识的本体。他们用来理解扩充用户的查询。他们通常在较大，但是不经常更新。这些本体一般都是进行横向发展，想要包含所有可能找到的概念，但并不描述细节。他通常包含的是对自然语言的一个有层次的描述(单词以及他们的意思)。这类本体中所用到的关系是较少的，通常有“近义词”，“反义词”，“上位(是一种)”(is-a)，“属性”(part of)等关系。相比较而言，动词包含的关系较多，因为需要有附加的关系来形容动作。而大多数概念型的表示都可以用形容词或者可作形容词的名称来形容。所以一般来讲，这种本体可以用一个较小的结构来进行表示。

自然语言本体定义了对用户的查询语句最基本的诠释，但他们必须与一些具体的术语以及具体的领域本体相关联。对于此种目的而言，一些比较有名的自然语言本体如 WordNet 并不能满足这样的需求。所以丰富自然语言本体的内容是一个重要的工作，而在这方面，研究界特别是针对汉语的研究中，中文自然语言本体本身就十分少，目前比较有名的开源资源只有东华大学直译过来的中文 WordNet，其丰富性不仅无法满足查询流程的需求，而且对中文语言本身的特性不是很契合。我们试图通过结合中文 WordNet 与百度百科，中文维基等百科网站的结构化或半结构化数据来构建这样的本体。而且自然语言本体往往不需要自动的频繁的更新。他们通常是由人力配合完成的。基于以上的特点，我们提出了我们自己的本体模型。

除了自然语言本体外，了解查询流程中的另外两个本体有助于理解我们本土的创建:

领域本体（domain ontology）是对领域概念的详细描述。这样的领域一般都是纵向

的，局限在一个小的范围中。通常这样的领域本体是通过人力创建的，但是通过阅读发现也有一些自动化的帮助提供给领域专家进行创建，不过这样的帮助大多是对那些经验并不是特别丰富的领域专家。

本体实例（ontology instance）主要就是展现网页上的一块知识，个人理解是语义网的基本单位。就像我们现在的万维网上 是由不同布局的 HTML 文档组成，以后的语义网将是由隶属于不同领域本体的实例组成。这些实例可以被当作网页，也有通往其它实例的链接（就像我们现在网上的超链接）。他们可以被自动的生成，并且会以较高的频率进行更新。

3. 本体的建立

本体的建立一般有三种途径：手工构建，复用现成本体，和使用半自动方法。

我们项目中的本体的建立分为如下阶段：

1) 首先，我们需要确定本体的范围。本体必须是特定领域的一种抽象，而选择其他抽象总是可能的。抽象中包含的内容取决于这个本体奖杯如何使用以及预期的扩展。所以我们需要首先确定四个内容：

1.本体将覆盖的领域：一般用途（general purpose）

2.使用本体的目的：进行知识查询

3.本体为何种类型的问题提供回答：实体多且关系复杂的语句

4.谁将使用和维护这个本体：专业人士以及有特殊爱好者

2) 其次，我们要考虑本体的复用，因为语义网在以十分快的速度在进行推广，那么在定义本体的时候，很少从零开始。我们可以考虑从第三方获得可用的本体，作为自己构建本体的一个起点。我们将在复用现成本体的部分详细讨论这方面的内容。

3)接着，我们需要列举在本体中可能出现的术语，即在该本体中期望出现的所有相关术语的一个非结构化列表。如，名词是类名的基础，动词或动词短语是属性名的基础。在我们的系统中，我们通过中文维基与百度百科自动获取名词，即 Entity。而动词我们通过固定词条中的信息栏中的属性加以提取，同时也配合人工操作来提取归类。

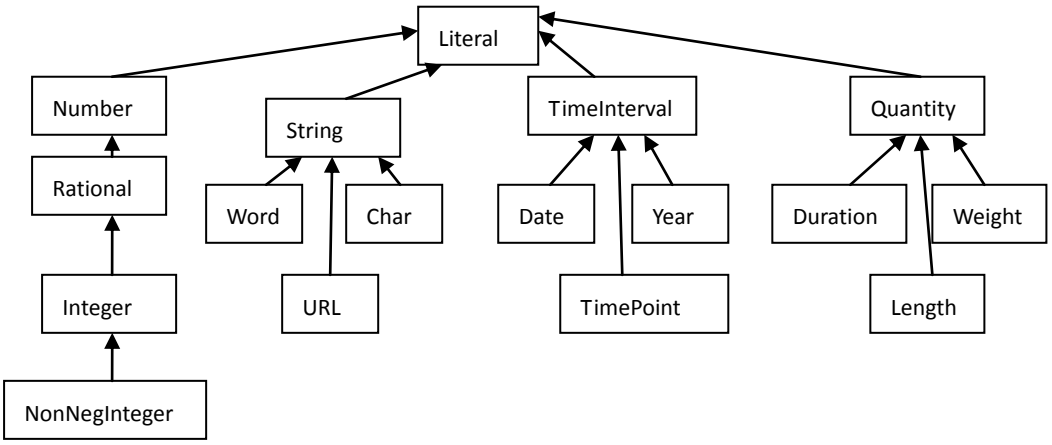
4) 然后，我们定义术语的分层结构，即他们的分类。我们采取的是以自顶向下的方式来做的这个工作。我们结合东南大学直译过来的中文 WordNet，利用其中的同义词集来找 subClassOf 这个关系。如果一个词是另一个词的下位词(hyponym),那么我们就认为前者跟后者有 subClassOf 这个关系。我们主要通过如下几个步骤：

1. 创建中文本体中上位词与下位词的关系。在上位词与下位词(hypernyms/hyponyms)创建连接的时候，我们首先用直译过来的 WordNet 找到中文词的 entry，然后用中文词的 ID 找到对应的英文的 ID，然后查询英文的上下位之类的词，最后在映射到中文上即可。
2. 将 WordNet 中的概念型的本体与中文维基和百度百科中较底层的实体相结合。在英文系统[Yago]中的做法是将一个词看为 Pre-modifier， head，还有 Post-modifier。然后看 WordNet 同义词集中有没有和 Pre+head 或 head 匹配的词。但是中文系统因为语言环境的不同需要采取不同的做法。对于出现了“的”字的中文维基分类（如歸化美國公民的德國人），我们用“的”作为标识将词语分开，取“的”字后的主体部分与 WordNet 同义词集进行匹配。对于一般的分类(如，20 世纪哲学家，德國猶太人等)，我们根据汉语的语言规则，发现主体部分一般都落于短语的后半部分。那么我们首先利用汉语的分词技术，将短语中的单词分开，然后取后半部分的单词（如，哲学家，犹太人）作为主体。

5) 接着，我们需要定义本体的属性。这一步通常与上一步交叉进行。在将一些类组织成层次结构时，自然要同时建立关联这些类的属性。举例而言，我们来看一下上面所列 subClassOf 关系的语义要求——只要 A 是 B 的一个子类，则对任何 B 实例成立的每一个属性陈述对任何 A 实例也都成立。由于这种继承的属性，我们可以把属性加在它所适用的最高层的类上。同时，在对类加属性时，应该立即陈述这些属性的定义域和值域。我们考虑了一般与具体的折中。一方面，给属性尽可能一般的定义域和值域，使之通过继承应用于子类。另一方面，尽可能窄地限定定义域和值域，有助于通过检查定义域和值域的违规来检测出本体中潜在的不一致性和概念偏差。

我们定义了如下规则:

1. 本体中的关系集都必须包含如下的关系：type, subClassOf, domain, range and subRelationOf
2. 本体中的实体集都必须包含如下的类：entity, class, relation and atr(用于检测无环传递性关系) ，并且还将包括以下的基本类:



6) 在上述几个步骤中，我们都只需要 RDF Schema 提供的表达能力，但不需要 OWL 增加的原语。原语是具有不可分割性的，即原语的执行必须是连续的，在执行过程中不允许被中断。但是，在我们定义属性侧面（facet）时，我们需要用到这方面内容，下面所说的在我们系统中大多都是利用 OWL 实现的。定义属性侧面是为了使属性定义更加完整，一般分为三个侧面：

1. 基数——为尽可能多的属性规定是否允许或必须有一定数目的不同值。比如说规定“至少有一个值”或“至多有两个值”等。在我们的系统中，比如说 isMarriedTo 这个关系，其最多就只能有一个值，因为一个人的配偶在同一时间只能有一个。
2. 特定值——经常用某个属性所具有的一些特殊值来定义类，这些特定值可以用 OWL 中的 owl:hasValue 加以指定。有时这个要求可以放宽，规定属性的值可以取自一个给定的类(owl:someValueFrom), 而不必是一个特定的值。
3. 关系特性——包括对称性，传递性，逆属性和函数值。我们定义了一种比较特殊的关系——非循环传递性的关系。比如 subclassOf 或 locatedIn。它定义了一种偏序的关系。

7) 在本体构建过程的这一步骤完成之后，就可以检查本体内在的不一致性了(在此之前无法完成这一步骤，因为 RDFS 不足以表达这些不一致性)。不一致性的常见情况有：

1）在传递属性，对称属性或逆属性的定义域和值域的定义中出现矛盾。

2）基数属性也常常是不一致性（比如萨科奇有好几任妻子，我们知识库中他与现任妻子布吕尼是 isMarriedTo 的关系，但是在检索网络文档时，我们常常会检测到“萨科奇与妻子塞西莉亚”之类的字眼，此时我们需要作不一致性检测）。

3）对属性值的约束与定义域和值域限定相冲突。

当然我们定义本体并不是因为要建立其自身的缘故，我们是想通过本体来组织集合实例。我们的集合实例取自中文维基及百度百科，将这些集合实例填入本体是一个独立的步骤，因为他们的数量比本体中类的数量搞很多个数量级，所以将他们置入本体无法通过手工完成，我们定义了一系列规则来自动抽取实例。

4. 本体语言的选择

本体语言用于对领域模型进行显示的形式化描述。其主要需求有良定义语法，高效率的推理支持，形式语义，充分的表达能力和表达的方便性。

良定义语法是机器处理信息的必要条件，通过在程序设计语言领域方面的阅读我们可以强烈地感受到他的重要性。基于 RDF 和 RDFS 构建的 DAML+OIL 和 OWL 都具有类似的语法。但是基于 XML 的 RDF 是否友好其实还有疑问，因为 OIL 等语法比其更适合人类阅读。目前终端用户是通过编辑工具即本体开发工具来建立本体，而不是直接通过 DAML+OIL 或 OWL。

形式语义能精确刻画知识的含义。他的含义不依赖主观直觉也不因人或机器而异。形式语义可以用来推导知识，对本体知识，可以用于推理类属关系，类等价，相容及分类。语义是推理支持的先决条件，上面的推导在语义的支持下可以用机器代替手工自动完成。推理支持的作用有：检查本体和知识的相容性，检查类间的隐含关系，对实例进行自动分类。

形式语义和推理支持的实现往往是通过把本体语言对应到已知的逻辑系统，并使用已有的自动推理机。OWL 部分地对应于描述逻辑，利用现有描述逻辑推理机如 FaCT 和 RACER。各种描述逻辑系统是谓词逻辑的一个具有高效率推理支持的子集。

我们在项目中之所以要用到如 OWL 的本体语言其实是因为 RDFS 对表达能力的局限性。“一般来说，表达能力越强的语言，推理的效率就越低，甚至是不可计算的。”[1]. 我们需要一个兼顾两方面需求的语言的语言，在能够高效率推理的同时又有描述各种本体知识的充分表达能力。理想情况下，OWL 应是 RDFS 的扩充，这样我们可以利用 RDFS 中类和属性的含义，然后增加建模原语以提供更强的表达能力。但因为 RDFS 中有这些强有力的建模原语(rdfs:class 表示所有类的类, rdf:Property 所有属性的类)，他们的表达能力很强，如果在这些原语的基础上扩展，则计算性能是无法保证的。

既然 RDFS 主要建模原语仅仅涉及以类型层次组织起来的词汇，包括子类关系和子属性关系，定义域和值域限定以及类实例。于是我们转而使用 OWL，在我们的项目中，我们主要用 OWL 实现以下的功能：

属性的局部限制，比如我要表现相对某些类的限制，但并不是将值域对所有类进行限定。比如吃这个概念，当主体是牛时，他的值域就是植物，但对肉食动物时，值域还可以包括肉。

类的不相交性。比如男性与女性不相交。但 RDFS 只能规定子类关系。

类的布尔组合。比如人类是男性和女性不相交的并。我们希望这种对现有类的布尔组合来产生新的类，但 RDFS 对此不支持。

基数约束。在前面提过，如一个人只有一个父亲或母亲，一门课只由一个或两个老师教授等。

属性的特殊属性。前面也提到过，偏序属性的传递等。

除了这些，随着 OWL 1.1 版的发布，其中提供了描述逻辑 SROIQ，其中有对非自反性和传递性的描述，这对我们项目中描述的非循环传递性的描述十分有帮助。

5. 逻辑规则的建立

前面我们主要介绍了知识表示相关的内容，包括关于网上资源内容的只是及相关领域的概念及其相互关系的知识(本体)。

早在万维网出现之前，逻辑特别是谓词逻辑一直是知识表示的基础。我们项目中所用的 RDF 和 OWL 可视为谓词逻辑的特殊形式。他们的公理语义与谓词逻辑十分相符。前面我们说了他们存在的理由之一是它们提供了一种能够很好地满足预定需求的语法。另一个理由就是他们定义了逻辑的可推理子集。OWL lite 和 OWL DL 粗略地对应于某种描述逻辑(Description Logic)，而这种描述逻辑是谓词逻辑的一个存在高效率证明系统(efficient proof systems)的子集，所以他们支持了高效率的推理，但是他们不能与 RDF 完全兼容(合法的 RDF 文档一般需通过一些扩展或限制才能成为合法的 OWL DL 文档)。

但是仅仅有描述逻辑是不够的。谓词逻辑的另一个存在高效率证明系统的子集包括规则系统(rule system),也称为 Horn 逻辑。其中定义了规则有如下形式:

A1, ……, An → B

对此，我们有两种解释：

1. 演绎规则(Deductive Rule)。如果 A1, … , An 已知为真，则 B 为真。
2. 反应式规则(Reactive Rule)。如果 A1, … , An 条件为真，那么执行行动 B。

规则系统是必要的因为他和描述逻辑是正交的，即他们不互相包含。那么他便能实现 OWL 所不能实现的功能。比如：OWL 无法表述 uncleOf(X, Z)这样一个关系，因为它需要对两个关系做链接操作：判断当 BrotherOf 通过 X 映射产生的值与 ChildOf 通过 Z 映射产生的值是一样的。这有点类似数据库中的 Join 操作，超出了 OWL 的范围。但规则可以很方便地进行表示：

brotherOf(X, Y), childOf(Z, Y) → uncleOf(X,Z)

当然，这些规则的判断有赖于我们对现实世界基本规则的收集。我们主要通过 WordNet 的同义词集还有中文维基和百度百科的转链接来得到。例如两个词“城镇中心”和“国际性大都市”，他们都属于同义词集“城市”。我们在本体中用 Means 来表示这样的关系：“ Metropolis ”Means City. 在集合实例到本体的阶段还有这样的规则： isPresidentOf(X,France) → Nationality(France, X)。

在这里，我们列举我们在本体中用到的一些基本规则(主要与 subRelationOf，acyclicTransitiveRelation (非循环传递关系)，type，range（值域），domain（定义域）有关):

(1) {(r1, subRelationOf, r2), (x, r1, y)} ,! (x, r2, y)

(2) {(r, type, acyclicTransitiveRelation), (x, r, y), (y, r, z)}

,! (x, r, z)

(3) {(r, domain, c), (x, r, c)} ,! (x, type, c)
(4) {(r, range, c), (x, r, y)} ,! (y, type, c)
(5) {(x, type, c1), (c1, subClassOf, c2)} ,! (x, type, c2)

这样的规则有很多，我们的规则基本都是单调规则，非单调规则我们一般都运用在“不知道就当不对”这种情形下，当获取了这方面的知识后会造成规则的不单调，这里涉及过多的技术细节，便不在这里详细讨论了。

6. 对时效性知识的处理

知识是有时效性的，比如说之前所举萨科奇的例子，我们经常会在网页上发现“萨科奇与妻子塞西莉亚”之类的字眼。但我们知道，这个萨科奇已经与塞西莉亚离婚了，他现在的妻子是布吕尼而不是塞西莉亚，这个知识是过期的了。那么我们如何判断知识是否过期呢？我们需要对陈述做出陈述。比如说我们想表述，爱因斯坦获得了诺贝尔奖这个事实是发生在 1921 年。

我们的想法是引入一个辅助对象，即 fact identifier，并通过主语(subject)，谓语(predicate)，和宾语(object)三个属性把它与原陈述的三个部分联系起来。如上述例子，我们就可以表述为：

#1 : AlbertEinstein hasWonPrize NobelPrize
#2 : #1 time 1921
其中#1 与#2 都是 fact identifier.

因为 RDF 只允许三元组，故无法直接给一个三元组增加一个标识符(那就成了四元组)，那么 RDF 如何满足我们对用 fact identifier 的需求呢？ 我们通过 RDF 的具体化机制(reification mechanism)达到这个目的。

RDF 的具体机制有两种表示方法，
一种是利用 rdf:Statement 的语法，如下所示：

```
<rdf:Description rdf:ID="item10245">
        <exterms:weight rdf:datatype="&xsd;decimal">2.4</exterms:weight>
    </rdf:Description>

    <rdf:Statement rdf:about="#triple12345">
        <rdf:subject rdf:resource="http://www.example.com/2002/04/products#item10245"/>
        <rdf:predicate rdf:resource="http://www.example.com/terms/weight"/>
        <rdf:object rdf:datatype="&xsd;decimal">2.4</rdf:object>

        <dc:creator rdf:resource="http://www.example.com/staffid/85740"/>
    </rdf:Statement>
```

这里直接用 rdf:subject，rdf:predicate 和 rdf:object 指明了主语谓语以及宾语。但是 triple12345 还是整个 RDF 图的一部分，他是图中单独分出的一块资源，而不是对整个图的概括。事实上 RDF 不提供内置的这种能够标识整个 RDF 的信息，这个标识只能作为图中一块特殊的资源与陈述关联。如果想要有一个标识能够概括整个 RDF 图，这种机制可能要用 RDF 以外的方法。

RDF 具体化的另一种方法是利用 rdf:ID 的属性，如下所示：

```
<rdf:Description rdf:ID="item10245">
      <exterms:weight rdf:ID="triple12345" rdf:datatype="&xsd;decimal">2.4
      </exterms:weight>
  </rdf:Description>

  <rdf:Description rdf:about="#triple12345">
      <dc:creator rdf:resource="http://www.example.com/staffid/85740"/>
  </rdf:Description>
```

其中用 rdf:ID="triple12345" 代表了这个事实，再在另外的 Description 中用#triple12345 作为主语来形容这个事实。

我们的本体主要采用后面一种表现方式，如前述“爱因斯坦获得了诺贝尔奖这个事实是发生在 1921 年”的例子。我们在 RDF 中可采取如下的表现方式。

```
<rdf:Description
rdf:about="http://stap.sjtu.edu.cn/semantic_ontology#Albert_Einstein">
<yago:bornInYear rdf:ID="f1">1879</yago:bornInYear>
</rdf:Description>
<rdf:Description
rdf:about=" http://stap.sjtu.edu.cn/semantic_ontology #f1">
<yago:foundIn rdf:ID="f2" rdf:resource="http:..."/>
</rdf:Description>
```

七．成果与评估

我们主要简单地对精确度和知识库的容量进行了评估。

精确度，我们用基本常识和人工检测来衡量我们本体的精确度。由于是我们自己的开发人员进行衡量，可能会带有一些主观的倾向性。我们随机抽取了一些事实然后让人们去判断是对还是错。如下是我们的结果：

Heuristic	#Eval	Precision
hasExpenses	46	100.00%
hasInation	25	100.00%
hasLaborForce	43	97.67441%
during	232	94.48950%
ConceptualCategory	59	92.94342%
participatedIn	59	91.94342%
plays	59	90.94342%
establishedInYear	57	90.84294%
...		
WordNetLinker	56	85.11911%
...		
InfoboxType	76	84.08927%
hasSuccessor	53	83.86150%
...		
hasGDPPPP	75	81.22189%
hasGini	62	81.00750%
discovered	84	80.98286%

如下是我们知识库的容量:

Relations	92
Classes	84,391
Individuals	531,588
People	146,308
Locations	80,988
Institutions/companies	17,893
Movies	10,234

八. 问题与展望

在中文本体的构建上我们还存在许多问题。

首先，中文没有一个规范开源的自然语言本体供大家使用与拓展。在我们项目中使用的中文 WordNet，事实上是先把我们的汉语词语翻译成 ENGLISH，然后在原有的英语 WordNet 里找到相关的上位词/下位词，在转成汉语显示出来的。其实要建立纯中文的本体也是有一定困难的，如果连描述诸如类与实例的关系等本体内部关联都使用中文似乎就已经超出了普通中文本体的概念范畴，可能通过 OWL 都无法描述，恐怕得再单独建立一个中文化的本体描述语言才行。自然语言本体是做语义搜索的第一步，它负责

将用户的查询定向到领域本体，并将查询语句规范模式化。一个符合中文语言规范的自然语言本体是十分必要的，而且它会极大地推动这个领域的发展

其次，在客观知识的来源方面，对中文的支持远远不及英文社区的支持。维基百科中文网站要达到英文网站那种品质还需要很长一段时间。而百度百科上的知识更是十分地不权威，任何互联网用户都可以对其中的很大一部分词条进行随意的篡改，页面上常常会出现“恶搞”的现象，以这种资源做知识搜索的来源实在令人无法放心。我们项目所选取的百度百科的页面，全为等级达到四级以上，通过率为 85%以上的百科专家进行撰写的。我们也通过编写人的相关数据对数据的可信度进行了加权。

最后，对于我们的项目还有改进之处。我们在构建本体的时候，由于客观原因的影响，大部分的精力花在了信息提取以及数据的预处理上面。因为没有一个规范的中文知识的来源，去除数据噪声的任务十分繁重。在后一阶段，我们主要将对本体可进行的应用开展研究。语义查询是由自然语言本体，与领域本体以及本体实例一起构成的。我们将会侧重于研究对知识的搜索以及对本体的查询上，我们已经开发了一小部分基于 SPARQL 以及 SPARQ2L 的应用；同时，我们还会对用户的查询语句进行一部分的自然语言分析。

参考文献

[1] A. Maedche. Ontology Learning for the Semantic Web. New York: Springer 2002.

[2]N. Noy and D. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. <http://www.ksl.stanford.edu /people /dlm/papers /ontology101 / ontology 101-noy-mcguinness .html>.

[3] M. Uschold and M. Gruninger. Ontologies: Principles, Methods and Applications. Knowledge Engineering Review, Volume 11, 2 (1996): 93 – 155.

[4] B. Omelayenko. Learning of Ontologies for the Web: The Analysis of Existing Approaches, In Proceedings of the International Workshop on Web Dynamics, 8th International Conference on Database Theory (ICDT 2001). 2001.

[5] Fabian Suchanek, Gjergji Kasneci, Gerhard Weikum: YAGO: A Large Ontology from Wikipedia and WordNet. Journal of Web Semantics, 2008

[6] Qi Zhang, Fabian M. Suchanek, Lihua Yue, Gerhard Weikum: TOB: Timely Ontologies for Business Relations. WebDB 2008

[7] Adar, E., Skinner, M., Weld, D.S.: Information Arbitrage across Multi-Lingual Wikipedia. WSDM 2009

[8] Ceri, S.: Search Computing. ICDE 2009

[9] Taneva, B., Kacimi, M.,Weikum, G.: Gathering and Ranking Photos of Named Entities with High Precision, High Recall, and Diversity. WSDM 2010i

[10] Wang, Y., Zhu, M., Qu, L., Spanioli, M., Weikum, G.: Timely YAGO: Harvesting, Querying, and Visualizing Temporal Knowledge from Wikipedia, Demo Paper. In: EDBT 2010 (2010)