

doi:10.3772/j.issn.1000-0135.2011.05.013

基于文本蕴涵的受限领域自动问答方法研究¹⁾

欧石燕

(南京大学信息管理系, 南京 210093)

摘要 本文提出了一种用于受限领域自动问答的新方法。与传统的自动问答方法不同,该方法不对用户问题进行语言分析以生成结构化的查询,而是利用文本蕴涵技术从自动生成的假设问题库中寻找用户问题所蕴涵的假设问题,然后利用该假设问题所附的答案指南获取用户问题的答案。该方法的优点之一是不需要复杂的语言处理就可以回答复杂的问题,而且正确率较高;优点之二是可以利用本体方便地生成不同语种的假设问题,实现跨语言问答。本文基于英文提问对该方法进行了测评,测评结果表明几乎所有的用户问题都可以基于生成的假设问题来直接或间接地回答,回答的正确率达到65.6%。

关键词 自动问答 文本蕴涵 本体

An Entailment-based Question Answering Method in a Restricted Domain

Ou Shiyan

(Department of Information Management, Nanjing University, Nanjing 210093)

Abstract This paper presents a new knowledge-based Question Answering (QA) method for a restricted domain with the use of textual entailment. In this method, a set of question patterns, called hypothesis questions, was automatically produced from a domain ontology, along with their corresponding query templates for answer retrieval. Then the QA task was reduced to the problem of looking for the hypothesis question that was entailed by a user question and taking its corresponding query template to produce a complete query for retrieving the answers from underlying knowledge bases. This method is able to answer complex questions without involving complex language processing and can be easily ported to other languages. An evaluation was carried out to assess the accuracy of the QA method, and the results revealed that almost all the user questions can be answered directly or indirectly based on the produced hypothesis questions, and the QA accuracy is 65.6% with the use of a semantic entailment engine enhanced by the domain ontology.

Keywords question answering, textual entailment, ontology

1 引言

自动问答(question answering, QA)是一种新的、智能化的信息检索技术,它允许用户以自然语言的方式精确地表达信息需求,系统则返回给用户符合其问题的特定答案,而不是一组可能包含答案的相

关网页。在自动问答技术的发展过程中,存在着两种并行发展的系统:基于知识的问答系统(knowledge-based QA)和基于自由文本(free text-based QA)的问答系统。基于知识的问答系统是从结构化数据中而不是自由文本中抽取答案,因此可用于处理更复杂的提问并获得更精确的答案^[1]。从早期的基于数据库的问答系统到近年来的基于本体

收稿日期:2010年3月29日

作者简介:欧石燕,女,1971年生,新加坡南洋理工大学博士,英国伍尔弗汉普顿大学博士后,现为南京大学信息管理系教授,主要研究方向:文本挖掘,语义网技术和数字图书馆。E-mail:ou.shiyan@gmail.com。

1) 本文受欧盟研究项目 QALL-ME (FP6 IST-033860 at <http://qallme.fbk.eu>) 的资助。

的问答系统,这种基于知识的问答系统的难点在于如何将一个自然语言提问精确地转换成为一个用某种查询语言(如 SQL、SPARQL)描述的结构化查询,从而能够使用该查询从底层的数据库或知识库中抽取正确的答案。这种转换是一个非常复杂的过程,需要对提问进行深层次的语言处理以构建它的逻辑表示,有时甚至需要手工介入。此外,由于语言的多样性,相同的问题往往可以采用多种不同的形式来表达,没有必要对这些含义相同的表达形式一一进行处理以获得相同或非常相近的查询。为了解决这一难题,Kouylekov 等提出了基于文本蕴涵的解决方案^[2]。文本蕴涵(textual entailment)是位于两个陈述性文本间的一个具有方向性的语义关系,其定义如下:

假设有两个文本: T (Text) 和 H (Hypothesis), 如果 H 的语义可以从 T 中推断出来, 那么则可以说 T 蕴涵了 H , 表示为 $T \rightarrow H$ 。

采用文本蕴涵识别技术可以判断看似不同的问题是否蕴涵了相同的语义,从而可以共享相同的查询方式。

本文提出了一种基于文本蕴涵的用于受限领域自动问答的新方法。与传统的自动问答方法不同,该方法不对问题本身进行语言处理以生成结构化的查询,而是先构建一组假设问题(hypothesis question, HQ)以及相对应的获取每个假设问题答案的指南(answer instruction, AI),然后通过文本蕴涵识别技术寻找用户问题(user question, UQ)所蕴涵的假设问题 HQ,最后利用该假设问题 HQ 所附的答案指南 AI 来获取用户问题 UQ 的答案。基于 Harabagiu 和 Hickl 的研究^[3](详见第 2 节)可以得出一个推论:如果用户问题 UQ 蕴涵了假设问题 HQ^①,那么假设问题 HQ 答案的某个子集也是用户问题 UQ 的答案。因此从某种程度上可以借助假设问题 HQ 的答案指南 AI 获取用户问题 UQ 的答案。该自动问答方法有两个关键点:一是如何从假设问题库中识别出用户问题所蕴涵的假设问题;二是如何生成假设问题及其答案指南。

在受限领域中,可以采用本体来描述领域知识,同时本体也限定了能够回答的用户问题的范围和范围。因此本文以旅游领域为例首次提出了利用领域本体自动生成假设问题的方法。这里生成的假设问题不是一个真正的问题,而是含有空槽的问题模板,其相应的答案指南是含有空槽的 SPARQL 查询模板。该方法的优点之一是不需复杂的语言处理就可

以回答复杂的用户问题,而且能够生成较精确的查询从而提高回答的正确率;优点之二是由于假设问题及获取其答案的查询是基于本体自动生成的,因此可以方便地生成不同语种的假设问题(如英文和中文),只要能够实现针对不同语种的文本蕴涵识别引擎,就可以轻松地实现跨语言问答。本文目前只实现了英文的文本蕴涵识别引擎,对中文的处理将在后续工作中进行,最终目的是实现中英双语种的自动问答系统。

本文的内容组织如下:第 2 节回顾了相关的研究工作;第 3 节介绍了旅游本体以及基于本体的知识库的构建;第 4 节介绍如何自动解析本体生成假设问题;第 5 节介绍基于文本蕴涵的自动问答流程;第 6 节报告对问答方法的评测及其结果;最后一节是结论和对后续工作的展望。

2 相关研究

在基于知识的自动问答中,如何将一个自然语言表示的提问转换成结构化的表示是众多研究者致力解决的一个难题。对于这个问题的研究可以追溯到早期的基于数据库的自动问答系统,主要是通过语言处理将自然语言提问转换为结构化的数据库查询,代表性的系统有 Chat-80^[4], PRECISE^[5] 和 START^[6]。在基于本体的问答系统中,因为领域本体的存在,可以依靠本体对自然语言提问进行语义注释,然后将它转换成为一个本体化的表示,如 MOSES^[7] 和 Aqualog^[8]。

近年来,文本蕴涵作为一种解决语言多样性难题的手段被运用到自然语言处理的许多任务当中^[9]。在自动问答任务中,一些研究者利用用户提问和一个答案已知(或获取答案的途径已知)的表达之间的蕴涵关系,使用该表达的答案(或获取答案的指南)来回答用户提问。采用这种方法的一个典型例子是参考文献[3]。在该文献中,Harabagiu 和 Hickl 认为“如果一个问题 Q 蕴涵了另一个问题 Q' ,那么 Q' 的部分答案也将是 Q 的答案”^[3]。因此,他们从信息检索系统返回的相关文本中自动生成一系列自然语言的问句及其答案,然后寻找用户提问所蕴涵的问句并将其答案作为用户问题的答案^[3]。

^① 因为文本蕴含关系被定义在两个陈述性文本之间,因此问句间的蕴含关系是指转换为陈述形式表达的问句间的蕴含关系。

另一个相类似的工作是参考文献[9]。Negri 等手工构建了一组关系模板并且每个模板附了一个答案指南,然后采用用户问题所蕴涵的关系模板的答案指南来获取用户问题的答案^[9]。

本文的研究与参考文献[3]和[9]相类似,但有所不同。不同之处主要在于:①本文生成的是问题模板及其答案指南(即查询模板)而不是如参考文献[3]中真正的问题和答案;②本文依靠本体自动生成假设问题及其答案指南,而不是如参考文献[9]中需手工构建关系模板及答案指南;③本文只生成基本类型的问题模板,对于复杂的用户问题可分解成几个简单的问题再与模板相匹配,而参考文献[9]中需要定义各种类型的关系模板,否则很多问题无法找到相匹配的模板。

3 领域本体和本体化的知识库

在本研究中,作者首先用 OWL-DL 语言构建了一个旅游领域的本体。建立该本体的目的是对旅游地的旅游资源和旅游设施以及在该地举办的与旅游相关的事件和活动进行概念化的定义和描述,提供一个旅游领域内的共享词汇表。本体中的每个类和属性都采用中英两种标签描述,

英文标签用于对英文问题的分析和处理,中文标签则是为中文问题服务。本体的主要结构如图1所示,其详细信息可参见参考文献[10],这里只做一个简单介绍。

该本体包含三个主类(main classes):

(1) Site:指为游客提供某种服务的地点,包括旅馆、饭店、旅游景点及旅游相关基础设施。

(2) Event:指与旅游相关的事件和活动,描述的是事件的静态信息,即事件的内容。

(3) Occurrence:指事件的动态信息,即某一事件在某一时间地点发生(如会议的召开)。

除以上主类外,该本体还包含几个从类,如 Facility、Period、Contact 等,这些从类依附于主类,用以描述主类的部件或属性。

在定义了领域本体之后,可以基于本体构建旅游信息知识库。作者开发了一个包装器,用于从旅游网站上抽取旅游信息,然后基于本体的类和属性对抽取的信息进行语义标注将其转换为 RDF 格式。构建的 RDF 知识库中含有结构化的旅游数据用以为用户问题提供答案,可采用 RDF 查询语言 SPARQL 从知识库中抽取特定的信息。

4 假设问题的生成

为了构建能够与用户问题相匹配的假设问题,需要了解用户问题的内容、类型和构成。因此从用户问题库中随机抽取了 500 个问题进行分析,结果见表1所示。

用户问题可分为三个主题:

(1) Site:查询有关地点的信息。

(2) Event:查询有关事件内容的静态信息。

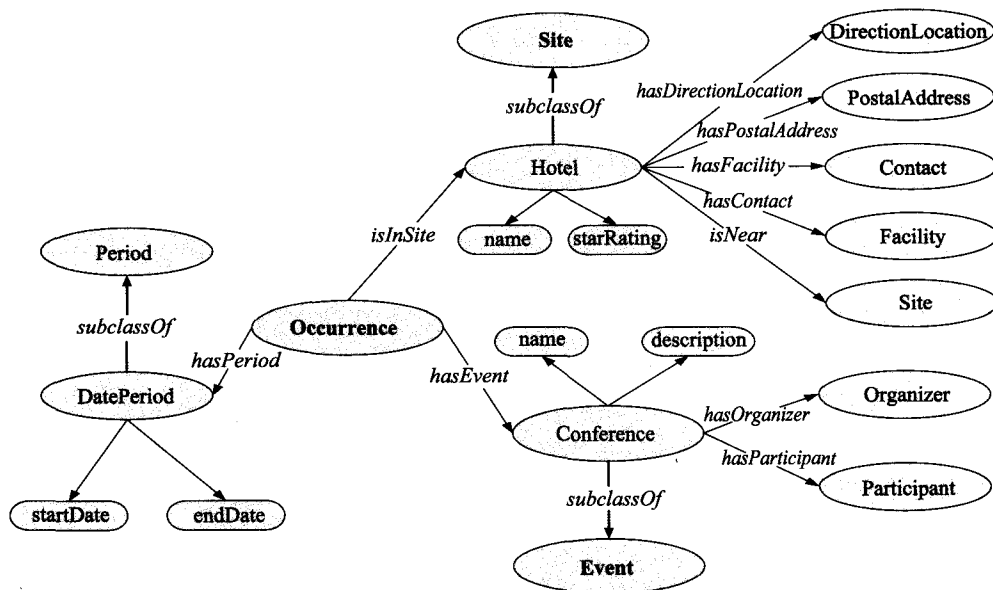


图1 旅游本体的主要结构图

(3) Occurrence: 查询有关事件发生的时间、地点等动态信息。

Site 和 Event 问题又可分为三种类型:

(1) T1: 查询符合某些条件的地点或事件的名称。

(2) T2: 查询某个名称已知的地点或事件的属性。

(3) T3: 查询符合某些条件且名称未知的地点或事件的属性。

表 1 用户问题的分析结果

主题	类型		用户问题范例
Site	T1 ^S		Please tell me the name of a Chinese restaurant in Walsall.
	T2		I want to know the address for the Kinnaree Thai Restaurant.
	T3		Could you give me a contact number for a modern Italian restaurant in Solihull?
Event	T1 ^S		What is the name of the film that stars Gerard Butler?
	T2		How long does the film Norbit last for?
	T3		When is the new Arnold Schwarzenegger film released?
Occurrence	Q_Event	T1 ^D	What is the name of the performance that is on at the Grand Theatre on 1st till 7th October
		T1 ^{D+S}	Could you tell me the name of an action movie which will be shown at Birmingham Showcase Cinema next week?
		T3	How long does the film last at Birmingham Showcase Cinema?
	Q_Site	T1 ^D	Can you tell me the name of the cinema showing Norbit now?
		T1 ^{D+S}	Please tell me the name of the cinema that is showing 300 and has disabled access.
		T3	How many seats are there for the cinema showing 300?
	Q_Period	T2	Please give me the dates that Christmas worship is being addressed at Saint Jude's Church?
		T3	Can you tell the time of the film that will be shown at Birmingham AMC Cinema next week?
	Q_Price	T2	What is the ticket price to see Aladdin at Birmingham AMC Cinema?
		T3	What is the price to see the movie that will be shown at 8:00PM at London AMC Cinema?

注:“S”表示该问题只含有静态信息,“D”表示该问题只含有动态信息,“D+S”表示该问题同时含有静态和动态信息。

T3 问题可以分解为一个 T1 问题和一个 T2 问题,譬如“请告诉我机场附近一家五星级酒店的地址?”可以分解为:

(1) T1: 请告诉我机场附近一家五星级酒店的名称?

(2) T2: 请告诉我<从 T1 中获得的酒店名>的地址?

Occurrence 问题可分为以下四个大类:

(1) Q_Event: 查询发生的事件的内容信息。

(2) Q_Site: 查询事件发生的地点信息。

(3) Q_Period: 查询事件发生的时间日期信息。

(4) Q_Price: 查询事件发生的价格信息。

上述每种大类又可进一步分为 T1、T2、T3 三种类型或其中的某两种类型。

根据以上分析发现,只需构建 T1 和 T2 两种假设问题,因为 T3 问题可以分解为以上两种问题。虽然理论上 T1 问题可以含有任意多个限制条件,但是假设问题中的限制条件最多为 3。原因有二:①研究表明 91% 的用户问题都只含有一至两个限制条件;②一个带有多个限制条件的问题可以分解为几个带有少量限制条件的问题。作者基于 Jena2^② 用 Java 语言实现了对 OWL 本体的自动解析,并以本体中的主类为出发点,对类的各种属性进行组合,共生成了 2703 个假设问题及其 SPARQL 答案查询模板。每个问题都具有中英两种描述,中文描述基于本体中类和属性的中文标签生成,英文描

② 一套开发语义网应用的 Java API。

述则基于英文标签生成。

以 Hotel 类为例,它的两个数据类型属性

“name”、“starRating”和一个对象类型属性“isNear”的组合,能够生成假设问题 H_1 :

What is the name of the hotel which has the star rating [Hotel. starRating] and is near the site [Site. name]? (具有[旅馆 星级]星级并位于[地点 名称]附近的一家旅馆的名称是什么?)

H_1 所附的 SPARQL 查询模板如图 2 所示。

```
@prefix xsd: http://www.w3.org/2001/XMLSchema#
@prefix qm: http://qallme.fbk.eu/ontology/qallme.owl#

SELECT ?hotelName
WHERE {
  ?Hotel qm:name ?hotelName.
  ?Hotel qm:starRating "[Hotel.starRating]"^^<xsd:int>.
  ?Hotel qm:isNear ?Site.
  ?Site qm:name "[Site.name]"^^<xsd:string> }
```

图 2 假设问题所带的 SPARQL 查询模板样例

5 自动问答流程

在构建完假设问题及其答案查询模板之后,自动问答就被简化成了一个文本蕴涵任务。整个自动问答流程如图 3 所示,主要包含四个步骤。

1) 命名实体和时间实体的识别

假设问题及其答案查询模板中含有空槽,这些空槽对应着蕴涵它的用户问题中的相应命名实体和时间实体。作者用本体的类和属性对用户问题中的命名实体(named entity, NE)进行注释,用 TIMEX2^[11]对时间实体进行注释。譬如对用户问题 U_1 的注释如下:

<U1> please tell me the name of a <NE type = "Hotel. starRating"> 3 </NE> star hotel that is near the <NE type = "Site. name"> President House </NE>. </U1>

2) 问题分类

为了减少假设问题的搜寻范围,可以先识别输入的用户问题的主题和类别,然后在相应的假设问题子类中寻找相匹配的假设问题。为此,作者采用朴素贝叶斯算法实现了两个问题分类器:一个是主题分类器,用以识别用户问题的主题,精确率为 95%;另一个是类别分类器,用以识别用户问题的类别,精确为是 85.5%。如果是一个 T3 问题,需要先对它进行分解后再输入到后面的文本蕴涵识别引擎中。

3) 文本蕴涵识别

用户问题 UQ 和假设问题 HQ 之间存在的蕴涵关系有三种:

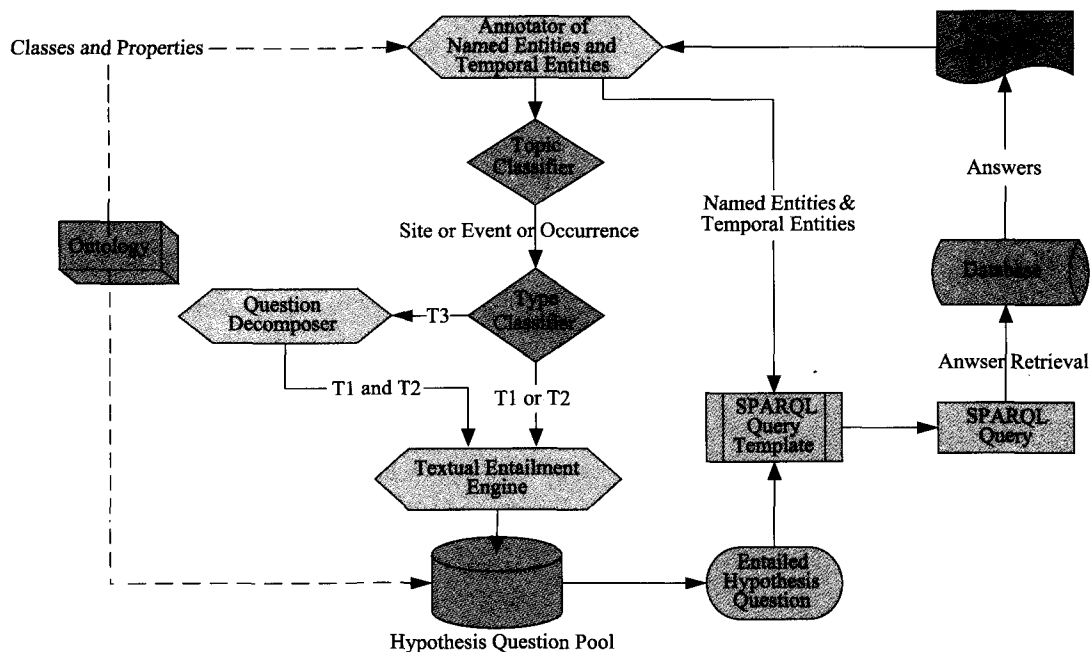


图 3 自动问答流程图

(1) 意译 (paraphrasing): UQ 和 HQ 看上去不同但实际上表达的含义相同, 譬如“Please tell me the name of a 3 - star hotel” = “What is the name of the hotel which has [Hotel. starRating = 3]”。

(2) 句法包含 (syntactic subsumption): UQ 中含有比 HQ 中更多的限制条件, 譬如“Please tell me the name of a 3 - star hotel in Beijing” → “What is the name of the hotel which has [Hotel. starRating = 3]”。

(3) 语义包含 (semantic subsumption): UQ 中含有比 HQ 中更具体的概念, 譬如“What is the name of the movie that will be shown at SAGA Cinema tomorrow” → “What is the name of the event that will occur in [Site. name = SAGA Cinema]”。

作者开发了两个文本蕴涵识别引擎来识别用户问题 UQ 和假设问题 HQ 间的蕴涵关系。一个是基于构词法的引擎 (syntactic engine), 采用最简单的“词袋法”, 通过两个问题间的余弦相似性来识别它们间的蕴涵关系; 另一个是基于语义的引擎 (semantic engine), 利用本体中类和属性的层次化结构以及 WordNet^③ 中的 synsets (同义词集合) 间的映射关系来识别一对 UQ 和 HQ 问题中的上位词、同义词和体用类型^④, 从而对基于余弦相似性的蕴涵识别方法进行增强。文本蕴涵识别引擎的实现参见参考文献 [12], 在此不做详述。蕴涵关系的强度可以用一个蕴涵函数 $e(T, H)$ 来测量, e 值越大, 说明蕴涵关系越强。经过多次试验, 作者选定了一个阈值 0.09, 如果低于该值, 则认为蕴涵关系不存在。

(4) 答案提取

在找到了用户问题所蕴涵的假设问题后, 需将识别的命名实体和时间实体填充到假设问题所附的答案查询模板的相应空槽中以生成完整的查询。譬如用 U_1 中的命名实体“3”填充图 2 中的 SPARQL 查询模板中的 [Hotel. starRating] 槽, 用命名实体“President House”填充 [Site. name] 槽, 就生成了一个完整的 SPARQL 查询, 用于从 RDF 知识库中提取 U_1 的答案。

6 问答方法的测评

在本研究中, 不需要直接测评问题答案的正确性, 而只需测评在何种程度上文本蕴涵识别引擎可以找到正确的假设问题, 从而采用正确地答案查询模板来生成正确的 SPARQL 查询以获取答案。这是因为基于知识的自动问答系统是从结构化数据中获

取答案, 答案的正确与否取决于是否生成了正确的查询而不是查询过程本身。此外, 本研究也对问题分类是否有助于文本蕴涵识别引擎发现正确的假设问题进行了测评。

该测评所用的样本问题是从用户问题库中随机抽取的另外 200 个问题。为了建立文本蕴涵识别的黄金标准, 作者手工从假设问题库中找出每个用户问题所蕴涵的假设问题。如果在库中不能找到相匹配的假设问题, 则标记该用户问题为“0”。一个用户问题可以在不同程度上蕴涵多个假设问题, 有的假设问题含有其所有的限制条件, 有的只含有其部分的限制条件。蕴涵程度可以用一个 0 ~ 1 之间的权重表示:

蕴涵权重 (w) = 假设问题 HQ 中含有的限制条件数 / 用户问题 UQ 中含有的限制条件数

譬如用户问题 U_1 在不同程度上蕴涵假设问题 H_1 、 H_2 和 H_3 :

- H_1 : what is the name of the hotel which has the star rating [Hotel. starRating] and is near the site [Site. name]? ($w = 1.0$)

- H_2 : what is the name of the hotel which has a star rating [Hotel. starRating]? ($w = 0.5$)

- H_3 : what is the name of the hotel which is near the site [Site. name]? ($w = 0.5$)

根据手工蕴涵识别, 作者发现: 在 200 个测试问题中, 所有的用户问题都能直接 (T_1 和 T_2) 或间接地 (T_3) 找到不同程度匹配的假设问题, 手工蕴涵识别的精确率^⑤达到 98.7%。这个值不是 100%, 是因为有个别用户问题只能找到部分匹配 (即 $w < 1$) 的假设问题。

在自动蕴涵识别中使用前文所述的两个蕴涵识别引擎, 即句法引擎和语义引擎。假设所有自动生成的假设问题的集合为 D , 相应的每个主题的假设问题子集为 D_{site} , D_{event} , $D_{occurrence}$, 每个类别的假设问题子集为 D_{T_1} 和 D_{T_2} 。对于每一个用户问题 UQ_i , 依次与每个假设问题 HQ_j ($HQ_j \in D$ 或 D 的某个子集) 相配对, 计算蕴涵函数 $e_{ij}(UQ_i, HQ_j)$ 的大小, 然后

③ WordNet 是由 Princeton 大学开发的一种基于认知语言学的英语词典。

④ 体用类型是指由名词引申出动词或形容词, 如从 director 引申出 direct。

⑤ 蕴涵识别精确率是用户问题蕴涵的假设问题的最大蕴涵权重之和除以问题的总数。

选择 e 值最大的假设问题作为 UQ_i 所蕴涵的假设问题。如果最大的 e 值仍小于 0.09, 则认为没有 UQ_i 所蕴涵的假设问题, 标记为“0”。除了测评蕴涵识别引擎的性能之外, 还需评估问题分类是否对蕴涵识别有帮助。因此共进行了三个实验:

(1) 从整个集合 D 中寻找假设问题。

(2) 先识别用户问题的主题, 然后从相应的子集 D_{site} , D_{event} 或 $D_{occurrence}$ 中寻找假设问题。

(3) 先识别用户问题的类别, 然后从相应的子集 D_{T1} 或 D_{T2} 中寻找假设问题。

如果没有采用问题分类, 则蕴涵识别精确率等于问答精确率; 如果采用了问题分类, 问答精确率则等于蕴涵识别精确率与分类器精确率的乘积。实验结果见表 2 所示。

实验结果表明, 语义蕴涵识别引擎的问答精确率最高可达 65.6%, 而句法引擎的问答精确率最高是 46.4%。通过先识别问题的主题, 可以缩小假设问题的搜寻范围, 从而提高文本蕴涵识别的精确率, 即使考虑到分类器的误差(5%), 两个引擎最后的问答精确率仍稍有提高(句法引擎从 42.5% 提高到 44.3%, 语义引擎从 65% 提高到 65.6%)。通过先识别问题的类别, 可以在很大程度上提高文本蕴涵识别的精确率, 尤其是对于句法引擎, 但是考虑到分类器的误差(14.5%), 最后语义引擎的问答精确率反而有所下降(从 65% 下降到 62.6%), 但是句法引擎的问答精确率有很大提高(从 42.5% 上升到 46.4%)。这说明问题分类对比较强大的语义引擎没有太大的帮助, 原因是语义引擎本身已经或多或少地利用了分类信息(如期望的问题答案类型)。但是对于性能不够好的句法引擎来说, 问题分类可以在很大程度上提高蕴涵识别精确率, 从而

提高最后的问答精确率。

7 结 论

本文提出了一种用于受限领域自动问答的新方法。该方法依靠领域本体预先构建问题模板(即假设问题), 然后依靠文本蕴涵识别来发现相应的问题模板用以回答问题。此方法非常适合于受限领域基于本体的问答系统, 但是也有不足之处: 一是生成的问题模板很难精确覆盖所有的用户问题, 尤其是对于含有较多类和属性的大型领域本体, 很难遍历所有的属性组合生成各种可能的假设问题; 二是对文本蕴涵识别引擎的依赖很大, 引擎的性能直接决定回答问题的正确率。

通过该自动问答方法进行测评表明: 生成的假设问题可以直接或间接地回答几乎全部的用户问题, 问答精确率达到了 65.6%。在理想情况下, 预先识别用户问题的类别可以减少假设问题的搜寻范围从而提高问答的精确度。但是实际中由于构建的分类器的精度还不够高, 因此问题分类只对比较弱的蕴涵引擎有帮助。

需要指出的是, 目前所用的蕴涵识别引擎的性能还不够高, 因而影响了该自动问答方法的正确率。在今后的工作中如何提高蕴涵识别引擎的精度将是研究的重点之一。目前作者只是着重处理了 T1 和 T2 问题。对于 T3 问题, 拟将开发一个问题分解器, 将 T3 问题分解成 T1 和 T2 问题后再进行处理。另外, 作者目前只实现了用于英文问题的文本蕴涵识别引擎, 在今后的工作中将开发中文蕴涵识别引擎, 这样可以实现中英双语种的跨语言问答。

表 2 蕴涵识别精确率和问答精确率

实验	主题或类别	用户问题数	假设问题总数	蕴涵识别精确率(%)		问答精确率(%)	
				句法引擎	语义引擎	句法引擎	语义引擎
1	All	200	2703	42.5	65.0	42.5	65.0
2	Site	122	2339	46.6	69.0	44.3	65.6
	Event	30	223				
	Occurrence	48	141				
3	T1	97	2458	54.3	73.2	46.4	62.6
	T2	85	245				
	T3	18	0				

注: 问题主题分类器的精确率是 95%, 问题类别分类器的精确率是 85.5%。

参 考 文 献

- [1] Mollá D, Vicedo J. Question answering in restricted domains: An overview [J]. Computational Linguistics, 2007, 33(1):41-61.
- [2] Kouylekov M, Negri M, Magnini B, et al. Towards entailment-based question answering: ITC-irst at CLEF 2006 [C]//Proceedings of the 7th Workshop of the Cross-Language Evaluation Forum. Heidelberg, Berlin: Springer, 2006:526-536.
- [3] Harabagiu S, Hickl A. Methods for using textual entailment in open-domain question answering [C]//Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL. Morristown, NJ: ACL, 2006:905-912.
- [4] Warren D, Pereira F. An efficient easily portable system for interpreting natural language queries [J]. Computational Linguistics, 1982, 8(3-4):110-122.2
- [5] Popescu A, Etzioni O, Kautz H. Towards a theory of natural language interfaces to databases [C]//Proceedings of the 8th International Conference on Intelligent User Interfaces. New York, NY: ACM, 2003:149-157.
- [6] Katz B, Felshin S, Yuret D, et al. Omnibase: Uniform access to heterogeneous data for question answering [C]//Proceeding of the 7th International Workshop on Applications of Natural Language to Information Systems. Heidelberg, Berlin: Springer, 2002:230-234.
- [7] Atzeni P, Basili R, Hansen D H, et al. Ontology-based question answering in a federation of university sites: the MOSES case study [C]//Proceedings of the 9th International Conference on Applications of Natural Language to Information Systems. Heidelberg, Berlin: Springer, 2004:413-420.
- [8] Lopez V, Pasin M, E Motta E. AquaLog: An ontology-portable question answering system for the semantic web [C]//Proceedings of the 2nd European Semantic Web Conference. Heidelberg, Berlin: Springer, 2005:546-562.
- [9] Negri M, Magnini B, Kouylekov M. Detecting expected answer relations through textual entailment [C]//Proceedings of the 9th International Conference on Intelligent Text Processing and Computational Linguistics. Heidelberg, Berlin: Springer, 2008:532-543.
- [10] Ou S, Pekar V, Orasan C, et al. Development and alignment of a domain-specific ontology for question answering [C]//Proceedings of the 6th Edition of the Language Resources and Evaluation Conference. Paris, France: European Language Resources Association, 2008.
- [11] Mani I, Wilson G, Sundheim B, et al. Guidelines for annotating temporal information [C]//Proceedings of the 1st International Conference on Human Language Technology Research. San Francisco: Morgan Kaufmann Publishers, 2001.
- [12] Ou S, Mekhaldi D, Orasan C. An ontology-based question answering method with the use of textual entailment [C]//Proceedings of the International Natural Language Processing and Knowledge Engineering. Menlo Park, California: AAAI Press, 2009: 212-219.

(责任编辑 马 兰)