文章编号:1001 - 9081(2003)09 - 0001 - 04

ALICE 机理分析与应用研究

夏 天,樊孝忠,刘 林
(北京理工大学 计算机科学与技术系,北京 100081)

摘 要:详细介绍了人工智能聊天机器人 ALICE 的知识组织结构和内部推理机制,指出它在中文处理方面存在的缺陷。在此基础之上,从分词与词性标注、同义句处理和 AIML 知识库的建设等多个方面探讨了 ALICE 在中文处理方面所面临的问题,并提出了相应的解决方法。

关键词:ALICE;AIML;模式;模板
中图分类号: TP391.2 **文献标识码**:A

ALICE Mechanism Analysis and Application Study

XIA Tian, FAN Xiao2zhong, LIU Lin
(*Department of Computer Science and Technology, Beijing Institute of Technology, Beijing* 100081, *China*)

Abstract: ALICE is an artificial linguistic Internet computer entity made by Lehigh University in USA. The knowledge organization and kernel deduction mechanism of it are introduced. There are some shortcomings when ALICE is used for Chinese processing. Relative solutions on word segmentation and tagging, synonymous sentence processing an construction of AIML knowledge dbase are proposed.

Key words: ALICE; Artificial Intelligence Markup Language (AIML); pattern; template

1 ALICE 简介

ALICE(Artificial Linguistic Internet Computer Entity),是由美国宾西法尼亚州 Lehigh 大学的 Richard S. Wallace 博士开发的一个基于经验的人工智能聊天机器人。它以 AIML 作为知识描述语言,目前存储了四万多条知识分类,因为其良好的设计和逼真的效果,在 2000 年和 2001 年两度获得著名的 Loebner 奖。

ALICE 的第一个版本于 1995 年完成,使用的是一种鲜为人知的基于集合论和数理逻辑的语言 SETL,起初并没有引起多少人的兴趣。1998 年,ALICE 用 Java 重新实现了 AIML 解析器,并把这个版本命名为 Program A。1999 年开发出的 Program B 版本,成了 ALICE 发展史上的一个重大突破,此时的 AIML 规范已经与 XML 语法完全兼容。2000 年 1 月,Program B 获得了著名的 Loebner 奖。同一年,Jacco Bikker 创造了第一个用 CPC ++ 语言实现的 AIML,称之为 Program C。Program B 虽然在很多平台上运行良好,但它有一个笨重的图形用户接口(GUI),更没有充分利用到 Java 2 的新特性,于是,Jon Baer 采用 Java 2 新技术重新编码了 Program B,并加入了许多新功能,形成了 ALICE 的 Program D 版本。自 2000 年 11 月起,Program D 便成了 ALICE 引擎唯一支持的 Java 版本。近来又出现了一系列用其它各种语言实现的 AIML 解析器,大都采用 AIML1.01 版本中所描述的规范。

ALICE 目前可以支持英语、德语以及法语对话,虽然 Wallace 博士说原理上也可以支持中文[1],但是由于汉语与西方语言差别很大,要让 ALICE 进行中文对话,还必须深入剖析它的运行机理,针对汉语的特点,做出进一步的改进和处理。

2 人工智能标记语言 AIML

ALICE 采用 AIML(Artificial Intelligence Markup Language) 作为它的知识描述语言。AIML 是利用 XML 标准定义的一种服务于人工智能领域需要的特定语言,设计 AIML 的最初意图就是为了能够用最简单的方式来创建人工智能聊天机器人,而且在语法上能与普通人所熟悉的 HTML 语法接近[2],然而 AIML 最早应用于 Program A 版本中时,并不符合 XML 语法。后来随着 XML 的发展,以及它所具有的灵活、易读、可扩展以及平台无关等优点,在 Program B 版本中,AIML 便已经与 XML 语法完全兼容。AIML 在整个聊天系统中的作用至关重要,可以说,ALICE 系统能否取得成功的关键,除了有一个好的推理机外,更重要的还是要看 AIML 格式的知识库建设的好坏,而且,ALICE 各种版本的核心实现也无非是围绕怎样方便、有效、快捷地组织和检索 AIML 知识分类这个问题进行的,因此,在论述 ALICE 内部推理机制之前,有必要对 AIML 做一个简单介绍。

2.1 AIML 语法构成要素

在 AIML 中,基本的知识单元是由分类(category)构成的,而每一个分类又是由用户输入的问题、ALICE 输出的答案和可选上下文环境(Optional Context)所组成[3]。一个简单的分类如下所示。

< category >

收稿日期:2003 - 03 - 18

作者简介:夏天(1978 -),男,山东潍坊人,博士研究生,主要研究方向:自然语言处理; 樊孝忠(1948 -),男,河南许昌人,教授,博士生导师,主要研究方向:自然语言处理、多媒体网络教学; 刘林(1974 -),男,辽宁人,博士研究生,主要研究方向:自然语言处理.

```
    <pattern> HELLO </pattern>
    <template> Hi, there! </template>
</category>
```

其中,模式 <pattern> 部分代表用户输入的问话,模板 <template> 部分则代表用户输入这一问句后,系统应该给出的答案。当然,AIML 还有其它许多重要标记,如递归调用标记 <srai> (Symbolic Reduction Artificial Intelligence),随机选择输出标记 <random> 等,下面给出了一个利用递归调用标记的例子:

```
<category>
    <pattern> HI </pattern>
    <template> <srai> HELLO </srai> </template>
</category>
```

对上面的两个例子来说,如果用户的问话为"HI",系统在找到对应的模板后,根据模板中的 <srai> 标记转而再以"HELLO"作为当前输入继续查找,最后找到"Hi,there!"作为最终结果返回用户。递归调用标记 <srai> 的使用在 AIML 中非常灵活,也非常重要,利用它可以把较长的句子转到长句的核心句,进行语法错误纠正,实现关键词匹配以及条件分支语句等等。AIML 规范中还定义了星号(*)和下划线(_)两个通配符,模式中出现星号或下划线的地方代表此处可以与一个到任意多个单词匹配成功,匹配时下划线的优先级最高。有关 AIML 的详细用法,可以参阅文献[4]。

2.2 AIML 知识库的结构

AIML 知识库由多个以 AIML 为后缀名的文件组成,每个 AIML 文件可以代表一个相似领域的可能话题,如关于地理方面的知识条目,我们就可以把它组织到 geography.aiml 文件中,以便于整体知识库的分类管理。一个简单的 AIML 文件内容如下所示:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<aiml version="1.0">
    <category>
        <pattern> HOW MANY DAYS IN WEEK </pattern>
        <template> 7 days per week. </template>
    </category>
    * *
    <category>
        <pattern> HOW MANY SECONDS IN YEAR </pattern>
        <template> Approximately 3.14 times 10 to the seventh.
        </template>
    </category>
</aiml>
* *
```

显然,一个 AIML 文件同时也是一个结构良好的 XML 文件,通过一个文字处理工具就可以修改、增删知识分类。

可以看出,AIML 就如同一个关于问题和答案的简单数据库,其中模式部分与 SQL 查询语言相似,但还要简单,由于模板中可能还包含递归调用标记,因此最终答案的输出并不仅仅依赖于第一个匹配上的分类,还与递归调用标记中的内容有关。

3 ALICE 内部推理机制

3.1 ALICE 系统工作流程

ALICE 系统工作流程如图 1 所示。

第一步:系统初始化

ALICE 系统在启动时,首先根据配置文件进行系统的初始化操作,把需要替换的词串(如把 itps 替换为 it is)、自身的相关信息(如名字、性别等)、人称转换信息以及此前的对话情景变量读入系统,并把 AIML 文件内容(即知识库)以树的结构形式加载到内存当中,形成内存知识树,这样当系统在响应用户输入的问句时,可直接在内存树中进行推理,提高了响应速度。加载完毕之后,等待用户输入问句。

第二步:接收用户输入,进行问句规范化处理

当 AIML 解析器接收到一行用户的输入后,首先把输入的文字分成单独的句子,进行问句规范化处理,分析当前句子中是否包含需要替换的字符串,如果有,则替换之。例如把问句中出现的"youpve"替换为"you have","Iam"替换为"I am"等等。问句规范化处理完以后,以规范问句到内存知识树中查询推理答案。

第三步:问句查询推理

这一过程是 ALICE 的核心部分,将规范化处理后的问句与内存知识树中的模式进行匹配,寻找最佳匹配结果,找到之后,读出该匹配模式对应的模板信息,进行下一步处理。

第四步:模板处理

也就是答案的后处理,模板中可能包含一些特殊标记需要处理,如读出机器人名字标记所代表的实际名称,还原星号部分所代表的内容,如果包含跳转标记,还需要在内存知识树中以跳转部分的内容做进一步的推理。

模板处理完后返回用户结果,等待用户输入新问句。

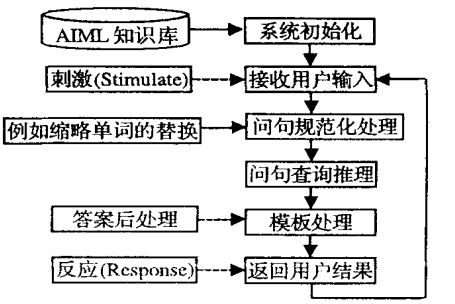

图 1 ALICE 系统工作流程

3.2 ALICE 的核心推理机制

ALICE 的核心推理部分称为 Graphmaster[1],它由一系列称为 Nodemapper 的节点集组成。每一个 Nodemapper 都有若干个从该节点出来的分支,这些分支可能是一个单词,也可能是一个通配符。如果加载官方公布的英文 AIML 知识库,Graphmaster 根节点的 Nodemapper 大约有 2 000 个分支,每一个分支代表一个模式的第一个单词。虽然 ALICE 存储了40 000 多条分类,但由于许多模式的第一个单词相同,这些相同的模式在 Nodemapper 中就指向了同一个子分支节点,所以即使是根节点,也只有 2 000 多个子分支而已,大大节省了内存空间。第二级分支节点的 Nodemapper 是模式句子中的第二个单词,它们的子分支指向紧跟在其后的第三个单词或者通配符,这样依次类推,直到子节点的分支到了模式句子的末尾为止。此时,把分类中该模式对应的模板内容存入该节点,这样 Graphmaster 就构成了一棵内存树,树的叶子节点存有模板信息,叶子节点的数目正好与知识库模式的数目相等。

根据用户的输入查找对应模式的过程就是 ALICE 的推理过程,假设用户输入的问句以单词 X 开头,那么首先拿 X 与 Graphmaster 中第一级节点的内容逐个比较,整个推理过程可

以分为三步:

1) 当前节点是否包含统配符"- "? 如果有,则搜索以当前节点为根节点的子树,以 X 后面剩余部分组成的子句在该子树中继续该过程,如果没有相匹配的节点,则转 2);

2) 当前节点是否包含 X? 如果是,则搜索以当前节点为根节点的子树,以 X 后面剩余部分组成的子句在该子树中继续该过程,如果未发现相匹配节点,则转 3);

3) 当前节点是否包含" 3 "? 如果是,则搜索以当前节点为根节点的子树,以 X 后面剩余部分组成的子句在该子树中继续该匹配过程。如果没有发现匹配节点,回溯到节点的父节点,把 X 重新加到子句子的首部,继续该匹配过程。

当进行匹配的句子到达句尾,并且匹配的最后一个节点包含模板内容时,则表明已经找到了相匹配的模式,此时,终止搜索匹配过程,返回该节点,取出模板内容,在必要信息处理完毕后把结果返回用户。由该算法可以看出,如果内存知识树中第一级节点含有关键字" 3 ",并且该节点含有模板信息,则对任何输入的问句,都能得到一个结果。

3.3　推理举例

下面举一个简单的例子来说明 ALICE 的实际推理过程。假设知识库中有以下三个分类:

```
<category>
    <pattern>- love your baby <Ppattern>
    <template> Yes, I love my baby. <Ptemplate>
<Pcategory>
<category>
    <pattern> do you love your family <Ppattern>
    <template> I love my family very much. <Ptemplate>
<Pcategory>
<category>
    <pattern> do you love 3 boy <Ppattern>
    <template> of course the boy is very clever <Ptemplate>
<Pcategory>
```

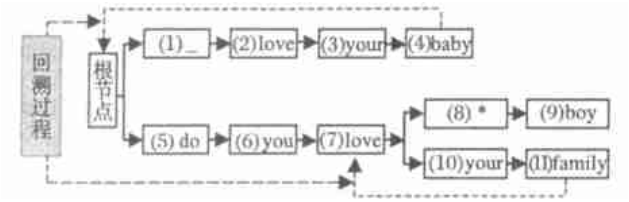

图 2　ALICE 内存知识树简图

针对这三个 category,给出它的内存知识树如图 2。为简明起见,图中节点只写出了它所包含的内容,滤掉了其它信息。假设用户输入的问句是" do you love your boy ",我们看一下它的搜索匹配过程。

根据图 2 和前面所讲的匹配规则,匹配串为" do you love your boy ",首先将问句的第一个单词" do "与第一级节点(节点(1)和节点(5))进行匹配,首先应该匹配节点(1),它仅有一个子节点(2),与问句下一个单词" you "匹配不成功,问句回退后" you "与下划线匹配成功,则继续匹配" love ",成功,匹配" your "成功,当再匹配" boy "时,当前节点(4)既不是"- ",也不是" boy "和" 3 ",匹配不成功,这样对节点(1)而言,第一步匹配下划线失败。继而执行匹配过程的第二步,匹配相同单词" do ",成功,匹配" you "成功,匹配" love ",与节点(7)成功,再匹配下一个单词" your "时,根据优先级,匹配下划线不成功,匹配原单词" your "成功,再匹配最后一个单词" boy "时,三个步骤都不成功,此时回溯到节点(7)。再以原单词" your "与节点(8)中的星号匹配,成功,继续匹配最后一个单词" boy ",与节点(9)匹配成功,问句结束,整个匹配过程完毕,叶子节点(9)为最后搜索匹配结果,其中也含有模板信息,表明找到对应了模式,在进行模板信息处理后,即可把结果返回给用户。

4　ALICE 在中文处理上的问题和解决方法

ALICE 总体上是一个基于经验的聊天机器人,它把它的经验知识以 AIML 的方式组织在一起,要想取得好的对话效果,必须预先提供大量的经验知识。当然,问话的方式是难于穷举完毕的,面对这个问题,规则与统计并用是最佳解决之道。因此 ALICE 系统也加入了一定的规则处理,在编写 AIML 时尽量把一些常见问句的形式抽取出来形成一定的规则,以降低编写的强度和规模,达到最佳效果。

ALICE 以词作为其处理的基本单位,在英语、法语、德语等屈折语对话方面取得了空前的成功,但是在处理汉语这样的孤立语方面,完全套用原来的构架显然不行:汉语词之间没有明显的分割标记,虚词运用较多,句序比较自由,ALICE 绝对严格的词与词匹配对汉语而言并不合适。因此要设计实现基于 ALICE 的中文问答机器人,必须针对汉语的特点,作进一步的研究和特定的处理。下面笔者从分词和词性标注、同义句处理、句式变换以及汉语知识库的建设四个方面对 ALICE 支持中文的问题进行了探索。

4.1　分词和词性标注

ALICE 在读取模式构建知识树和读取用户问句进行推理时,都是先把句子分成一个一个的词,再进行下一步处理的。像英语这样的西方语言是拼音文字,词之间以空格为分割标志,什么是词是明确的,偶尔出现的一些单词缩写和特殊标点符号的处理也比较简单,但是以汉字作为最小单位的汉语句子中,词与词之间没有明显的分割标志[4],因此,要处理汉语句子,就必须先作分词处理。

目前,在 NLP 领域中,汉语的自动分词已经有多种算法实现,大体上分为如下几种:最大匹配法、反向最大匹配法、逐词遍历法、设立切分标志法、最佳匹配法、有穷多层次列举法、二次扫描法、基于词频统计的分词方法、基于期望的分词方法、双向扫描法、邻接约束方法、专家系统方法、最少分词词频选择方法、神经网络方法等等,技术相对比较成熟。问题的关键是在什么地方引入分词处理,才能使 ALICE 既支持中文问答,又不破坏其原有的基本推理框架?我们所采取的方法是:当 ALICE 加载 AIML 知识时,首先对读出的模式进行分词处理,形成分词以后的模式句子,再交给 ALICE 的下一个模块继续处理,最终形成以汉语词语为节点的内存知识树。之后,当用户输入问句时,采取同样的分词策略对问句进行分词处理形成新问句,再到内存知识树中去推理查找相对应的答案模板。

在 NLP 处理过程中,分词和词性标注的工作是密切联系在一起的。在 ALICE 中,为了提高近似句子匹配成功的可能性,我们把词性标注的信息带入系统之中,在进行比较时,针对汉语的特点,对于一些非关键的成份,如部分助词、语气词等,如果没有匹配成功,不是简单地当作匹配失败来处理,而是跳过这些部分,继续余下部分的比较,以提高近似句子推理成功的比例。

4.2 同义句处理

即使是一个受限领域穷举可能出现的问句也是比较困难的,因此怎样针对一个已经存在的问句模式,判断出其它句子与它之间的相似性,也就是同义句的判定问题,就显得格外重要了。ALICE 在面对这个问题时,一方面是简单地把一部分词作了替换处理,如遇到"canpt"这样一个串,就把它替换成"can not",再继续正常处理;另一方面,如前面所述,可以在问句模式中加入通配符,以匹配一个或者一个以上任意多个输入符号串中的词,部分实现关键词匹配的效果;第三,在编写 AIML 知识条目时,可以做一定的句式变换,利用递归跳转标记,把多个意义相同的问句模式转到同一个模板。对英语而言,这三种方法已经能够解决很大一部分问题,可是对汉语而言,这还远远不够。

比如拿一个简单的句子"计算机是什么"为例,在保持基本语序不变的情况下,仍然可以有多种不同的说法,如:"计算机是啥","电脑是什么","电脑是啥"等等,在编写这一类知识时,原有的方法是列举出尽可能多的相似句子,利用递归跳转标记跳转到相同的答案模板,编写时要想保证不重复、不遗漏很困难。为此,我们扩展了模式原来的编写方法,把可能一起出现的成分用中括号括起来,各部分之间再用竖线隔开,加载知识时,先还原这类模式所有可能的组合句子,再把各句子分词加载。如前面所说的这个句子,就可以这样编写:[计算机| 电脑| 微机| 微型计算机]是[什么| 啥]。

当然,单纯依靠在编写知识时把所有的同义词枚举出来是不现实的,这样也会导致知识树中节点过多,耗费内存,降低速度。另外,汉语虚词运用较多,许多成分在句子中可有可无,要把这些可有可无的成分的各种组合一一列举出来写入知识库,是不可能的,也没有必要。针对这些问题,我们对 ALICE 的推理机制作了改进,利用董振东先生的知网资源,在 ALICE 推理过程中加入了词语相似度处理和打分机制,当两个词语进行比较时,不是只有相同和不同两种可能,而是计算它们之间的相似度,如果相似度超过一个阈值(比如大于 0.85),则认为它们匹配成功,可以继续余下部分的比较,当低于此阈值时,则看该词语在句子中是否可以过滤,如能过滤,继续匹配过程。同时我们对关键词、同义词和星号的匹配分别赋予不同的权值,对过滤的成分也给予一定的分数,当接收到一个问句时,首先把知识树中所有可能与之等价的一组模式选择出来,再根据模式与问句匹配时得到的权值之和扣除问句忽略部分的估计分数为评价标准进行择优,挑出最佳模板作为最终结果。当然,这种方式是以降低系统的速度和准确性为代价的,各个参数的取值也需要根据实际使用情况不断地调整。

4.3 句式变换

中文问话的方式虽然灵活多变,但仍有一定的规律可循,我们可以充分利用 AIML 中的递归跳转标记,总结常见问句抽取共同规则,作一定的句式变换,实现知识体与问句的形式相分离,从而更有利于知识的表示和收集,提高系统的推理效率。下面举例说明:

```
<category>
  <pattern> 3 在[哪里| 哪个位置| 什么地方] </pattern>
  <template>
    <think> <set name="search">位置</set> </think>
    <srai> <starP> </srai>
  </template>
</category>
<category>
  <pattern> 3 的首都在[哪里| 哪个位置| 什么地方]
  </pattern>
  <template>
    <think> <set name="search">首都位置</set>
    </think>
    <srai> <starP> </srai>
  </template>
</category>
<category>
  <pattern> 北京 </pattern>
  <template>
    <condition>
      <li name="search" value="位置">
        北京在中国的北部. </li>
      <li name="search" value="所属国家">
        北京是中国的首都,当然在中国. </li>
      <li> 北京是中国的首都,是一个国际化大都市.
        </li>
    </condition>
  </template>
</category>
```

其中,前两个分类可看作是问句规则,后一个可看作是关于北京的知识体。拿第一个分类来说,<think>和<set>表示推理到此处时,在内存中动态设置了一个名为"search"、值为"位置"的变量,<starP>标记表示该模式星号部分所代表的实际内容。第三个分类中的<condition>表示根据<li>中的变量值来选择其中的一项条目,如果变量"search"在内存中的当前值为"位置",就选择第一条作为结果输出,当所有条目中所指明的变量值都与内存中该变量的当前值不一致时,则选择不带变量的那项<li>的内容作为结果。

假设用户输入的问句是"北京的首都在哪里",该句子首先与模式"3 的首都在哪里"匹配成功,处理对应的模板时,把变量"search"的当前值设为"首都位置",此时星号代表北京,再以"北京"作为输入查找,与模式"北京"匹配成功,因为<li>中指明的 search 值与当前值"首都位置"不同,所以选择最后一项"北京是中国的首都,是一个国际化大都市"输出。

利用这种方式,我们就实现了问句方式与知识体的分离,比如要增加"美国"这一知识时,对美国的各种提问方式就无需再重复编写了。同样,利用句式变换,还可以对问句作前后缀过滤,把句子前面的客气语、后面的语气词等可有可无的成分滤掉,转而以核心句子到知识树中再查找。这就使知识的组织变得更加合理有效,同时也降低了知识编写的难度和系统的时空复杂度,如针对某领域内问句的方式有 m 种,知识体有 n 个,原来需编写 $m \times n$ 条知识,而现在只需 $m + n$ 条即可。

4.4 汉语 AIML 知识库的建设

ALICE 会话效果的好坏,与 AIML 知识库建设的规模和知识的覆盖面至关重要,目前,有关英文、法文方面的 AIML 知识库已经初具规模,亚洲印度尼西亚文方面的 AIML 也可以在 ALICE 网站上下载得到,然而,针对中文编写的 AIML,目前尚未发现相关资料,其中的原因是多方面的:首先,ALICE 的最初设计就是针对西方文字的特点进行的,处理过程中没有汉语分词等技术障碍;其次,汉语与印欧语系的语言差别很大,语序比较自由,虚词运用较多,大量可有可无的句子成分对于AIML的编写非常不利;再次,单凭个人的力量很

文章编号:1001 - 9081(2003)09 - 0005 - 03

Web 集群系统 QoS 模式的预测与分类

陈志刚 ,刘安丰 ,许 伟
(中南大学 信息科学与工程学院 ,湖南 长沙 410083)

摘 要:提出了一种线性时间序列与神经网络非线性时间序列相结合的"两级模式识别"的分级预测与分类的技术:首先采用线性时间序列分析方法对 Web 集群的服务模式进行总体预测与分类,在总体模式预测与分类的基础上采用神经网络非线性算法进行 QoS 模式精确识别。理论与实验分析结果证实了该方法的有效性。

关键词:QoS 模式;时间序列;神经网络;预测与分类
中图分类号: TP393.07 **文献标识码**:A

Prediction and Classification of QoS Pattern Based on Web Cluster

CHEN Zhi2gang ,LIU An2feng ,XU Wei
(*College of Information Science and Engineering , Central South University , Changsha Hunan* 410083 , *China*)

Abstract: The article introduces a level predicting and classifying method named "two level pattern recognization" , combining the linear time series and the nonlinear time series of neural network. We can predict and classify overally the service pattern of Web clusters by linear time series , then recognize exactly the QoS pattern by nonlinear algorithm of neural network. The method has been proved effective by theoretic analysis and experiments.

Key words: QoS pattern; time series; neural network; predict and classification

1 引言

服务质量(Quality of Service , QoS)控制技术作为下一代网络的核心技术之一,近年来一直是计算机网络中研究与开发的热点问题。如何对 Web QoS 进行分类与管理在国外有一些研究[1],但是在国内基本上还是空白。特别是在 Web 集群这样一个复杂系统中,它具有不同的应用和动态变化的客户请求服务,不同的应用和请求有不同的 QoS 要求,需要采取不同的控制策略。而进行 QoS 控制的基础就是对 QoS 模式的正确分类和识别,只有对确定的 QoS 模式采取相应的控制策略才是有效的。

前人已经在 QoS 控制[2]上做了大量的研究且富有成果。但是对于 Web 集群系统,由于应用系统的不同,其 QoS 控制策略与方法必然不一致,有的甚至是对立的。例如:在一个提供新闻娱乐、电子商务和教育内容三种类型服务的集群系统中,当这些服务要求冲突时,如何进行"QoS 协商"。文献[3]指出晚上 8 时,电子商务是用户请求最多且成交率最高的服务,那么按"最大回报率"原则,应最先满足电子商务的 QoS 需求。但是这种需求又是变化的,新闻娱乐也起着吸引客户的作用,所以对于新闻娱乐的服务质量也不能无条件的"区分服务"。特别是在当电子商务服务很少时,没有必要预留太多的资源而损害其它服务,降低系统利用率,这就需要"QoS 协商"。

收稿日期:2003 - 03 - 10;**修订日期**:2003 - 06 - 03 **基金项目**:湖南省自然科学基金项目(02JJY2097)
作者简介:陈志刚(1964 -),男,湖南长沙人,教授,博士生导师,主要研究方向:网络计算、分布式处理; 刘安丰(1971 -),男,湖南长沙人,博士研究生,主要研究方向:分布式系统、Web QoS; 许伟(1973 -),男,湖南长沙人,硕士研究生,主要研究方向:分布式系统.

难在短时间内建设好一个具有一定规模的 AIML 知识库,借助于互联网,实现网上知识收集和整理,将是一条比较可行的途径。目前,北京理工大学 NLP 实验室在这一方面作出了自己的尝试,初步建成了一个涉及日常会话、地理知识、银行业务咨询等 5 个领域,共计 3820 条分类的知识库,与同时进行的自动问答系统相结合,取得了较好的效果。

5 结束语

针对汉语的特点,对 ALICE 进行改造,设计实现支持中文对话的聊天机器人,在网络教学、Web 服务、呼叫中心、网络导航等应用中都有着广阔的应用前景[5]。同时,对于如何更加有效地获取、组织和检索知识,怎样对 AIML 进行合理扩充以增强它的知识表达能力,例如是否可以在模板中引入逻辑判断、怎样对通配符所代表的内容进行一定的限制等,都值得我们作进一步深入的研究。

参考文献

[1] Wallace RS. The Anatomy of ALICE[EB/OL]. http://www.alicebot.org/anatomy.html,2001.
[2] Wallace RS. Dont Read Me [EB/OL]. http://alicebot.org/articles/wallace/dont.html,2000 - 09.
[3] 戴开宇,张申生,王淼. 分布式虚拟环境中的聊天机器人的研究[J]. 计算机工程与应用,2002,(7):13 - 16.
[4] Wallace RS. AIML Overview [EB/OL]. http://www.pandorabots.com/pandora/pics/wallaceaimltutorial.html,2001.
[5] 赵铁军. 机器翻译原理[M]. 哈尔滨:哈尔滨工业大学出版社,2000.