



(12) 发明专利



(10) 授权公告号 CN 111149107 B

(45) 授权公告日 2023. 08. 22

(21) 申请号 201880062963.0

(22) 申请日 2018.09.28

(65) 同一申请的已公布的文献号
申请公布号 CN 111149107 A

(43) 申请公布日 2020.05.12

(30) 优先权数据
62/564,868 2017.09.28 US

(85) PCT国际申请进入国家阶段日
2020.03.27

(86) PCT国际申请的申请数据
PCT/US2018/053392 2018.09.28

(87) PCT国际申请的公布数据
W02019/067878 EN 2019.04.04

(73) 专利权人 甲骨文国际公司
地址 美国加利福尼亚

(72) 发明人 B·加里斯基 V·韦氏诺一 徐新

(74) 专利代理机构 中国贸促会专利商标事务所
有限公司 11038

专利代理师 周衡威

(51) Int.Cl.
G06F 40/205 (2020.01)
G10L 15/18 (2013.01)

(56) 对比文件
CN 101118554 A, 2008.02.06
CN 103871286 A, 2014.06.18
CN 106462604 A, 2017.02.22
CN 106558307 A, 2017.04.05
CN 106796787 A, 2017.05.31
US 2004148154 A1, 2004.07.29
US 2016055240 A1, 2016.02.25
US 8700404 B1, 2014.04.15
Michael Collins 等. New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. 《ACL02》. 2002, 263-270.

审查员 王咏冬

权利要求书3页 说明书27页 附图9页

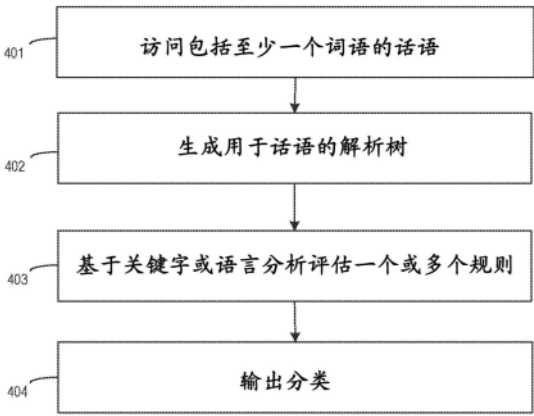
(54) 发明名称

使自主代理能够区分问题和请求

(57) 摘要

本发明的系统、设备和方法涉及文本分类。文本分类系统访问文本的话语。话语包括至少一个词语。文本分类系统生成用于话语的解析树。解析树包括具有词语类型的至少一个终端节点。终端节点表示话语的词语。文本分类系统将一个或多个规则应用于文本。然后，文本分类系统将话语分类为问题或让自主代理执行动作的请求。

400



1. 一种将文本分类为问题或请求的系统,包括:

语言引擎,被配置为:

访问包含至少一个词语的话语;

生成用于所述话语的解析树,其中所述解析树包括包含词语类型的至少一个终端节点,其中每个终端节点表示所述话语的词语或短语;以及

将包含一个或多个词语类型的一个或多个语言模板应用于所述解析树;以及

规则引擎,被配置为生成从所述语言引擎获得的一个或多个匹配结果,并通过应用包括以下各项的规则来确定所述文本的分类:

(i) 响应于确定语言模板匹配,将所述话语分类为请求,

(ii) 响应于确定所述话语包括祈使动词作为所述话语的第一词语,将所述话语分类为请求,

(iii) 响应于在所述话语中识别出一个或多个预定义的请求关键字,将所述话语分类为请求,以及

(iv) 响应于在所述话语中识别一个或多个预定义的问题关键字,将所述话语分类为问题,

其中,所述系统还基于所述分类被配置为向用户设备发送消息或调整外部设备的配置。

2. 如权利要求1所述的系统,其中,所述规则还包括:(v) 响应于未能将所述话语识别为问题或请求而将所述话语分类为未知。

3. 如权利要求1所述的系统,其中,所述语言模板包括第一人称代词和名词。

4. 如权利要求1所述的系统,其中,所述规则还包括:(v) 响应于确定所述话语在第一词语位置中包括问题前缀而将所述话语分类为问题。

5. 如权利要求1所述的系统,其中,所述一个或多个预定义的问题词语包括心理动词。

6. 如权利要求1所述的系统,其中,所述规则还包括:(v) 响应于确定所述话语在所述话语的最后词语位置处包括请求后缀而将所述话语分类为请求。

7. 如权利要求1所述的系统,其中,所述规则还包括:(v) 响应于确定所述话语包括事务动词而将所述话语分类为请求。

8. 如权利要求1所述的系统,其中,所述语言引擎被配置为在确定所述解析树之前从所述话语中移除一个或多个停用词。

9. 一种使用分类模型将话语分类为问题或请求的方法,其中问题寻求的是答案,并且请求寻求的是事务,所述方法包括:

访问文本的话语,其中所述话语包括至少一个词语;

生成用于所述话语的解析树,其中所述解析树包括包含词语类型的至少一个终端节点,其中每个终端节点表示所述话语的词语或短语;

通过将分类模型应用于所述解析树来确定所述话语的分类,所述应用包括:

为所述解析树确定第一相似度分数,所述第一相似度分数指示所述话语和被识别为问题的训练类别之间的第一匹配,以及

为所述解析树确定第二相似度分数,所述第二相似度分数指示所述话语和被识别为请求的训练类别之间的第二匹配;

从所述分类模型接收所述话语的分类,其中所述话语的所述分类(i)基于所确定的类别的相似度分数高于其它相似度分数,或者(ii)是未知的。

10.如权利要求9所述的方法,还包括:

访问包括训练数据对集合的训练数据集合,其中每个训练数据对包括文本和预期分类,并且其中所述训练数据集合包括(i)包含问题的第一训练数据对以及(ii)包含请求的第二训练数据对二者;以及

通过迭代以下操作来训练分类模型:

将训练数据对之一提供给所述分类模型;

从所述分类模型接收所确定的分类;

通过计算所确定的分类与所述预期分类之间的差来计算损失函数;以及

调整所述分类模型的内部参数以最小化所述损失函数。

11.如权利要求9所述的方法,其中,所述分类是包括与所述话语的最高相似度的分类。

12.一种将话语分类为问题或请求的方法,其中问题寻求的是答案,并且请求寻求的是事务,所述方法包括:

访问文本的话语,其中所述话语包括至少一个词语;

生成用于所述话语的解析树,其中所述解析树包括包含词语类型的至少一个终端节点,其中每个终端节点表示所述话语的词语;以及

执行来自操作集合中的一个或多个操作,所述操作集合包括:

(i)响应于确定所述解析树的第一终端节点包括祈使动词,

将所述话语分类为请求,以及

(ii)响应于在所述解析树中分类出一个或多个预定义的问题关键字,将所述话语分类为问题。

13.如权利要求12所述的方法,其中,所述操作还包括:

将包含一个或多个词语类型的一个或多个语言模板应用于所述解析树;以及

响应于确定语言模板匹配,将所述话语分类为请求。

14.如权利要求12所述的方法,其中,所述词语类型包括名词以及第一人称时态中的代词。

15.如权利要求12所述的方法,其中,所述操作还包括:响应于未能将所述话语识别为问题或请求,将所述话语分类为未知。

16.如权利要求12所述的方法,其中,所述操作还包括:(iii)响应于在所述话语中分类出一个或多个预定义的请求关键词,将所述话语分类为请求。

17.如权利要求12所述的方法,其中,所述操作还包括:(iii)确定所述话语包括问题前缀,将所述话语分类为问题。

18.如权利要求12所述的方法,其中,所述一个或多个预定义的问题词语包括心理动词。

19.如权利要求12所述的方法,其中,所述一个或多个预定义的请求关键字包括事务后缀或事务动词。

20.一种用于对话语进行分类的系统,包括:

存储非暂态计算机可执行程序指令的计算机可读介质;以及

通信地耦合到所述计算机可读介质以执行所述非暂态计算机可执行程序指令的处理设备,其中执行所述非暂态计算机可执行程序指令将处理设备配置为执行包括以下各项的操作:

访问文本的话语,其中所述话语包括至少一个词语;

生成用于所述话语的解析树,其中所述解析树包括包含词语类型的至少一个终端节点,其中每个终端节点表示所述话语的词语或短语;

通过将分类模型应用于所述解析树来确定所述话语的分类,所述应用包括:

为所述解析树确定第一相似度分数,所述第一相似度分数指示所述话语和被识别为问题的训练类别之间的第一匹配,

其中问题寻求的是答案,以及

为所述解析树确定第二相似度分数,所述第二相似度分数指示所述话语和被识别为请求的训练类别之间的第二匹配,

其中请求寻求的是事务;

从所述分类模型接收所述话语的分类,其中所述话语的所述分类(i)基于所确定的类别的相似度分数高于其它相似度分数,或者(ii)是未知的。

21.如权利要求20所述的系统,所述操作还包括:

访问包括训练数据对集合的训练数据集,其中每个训练数据对包括文本和预期分类,并且其中所述训练数据集包括(i)包含问题的第一训练数据对以及(ii)包含请求的第二训练数据对二者;以及

通过迭代以下操作来训练分类模型:

将训练数据对之一提供给所述分类模型;

从所述分类模型接收所确定的分类;

通过计算所确定的分类与所述预期分类之间的差来计算损失函数;以及

调整所述分类模型的内部参数以最小化所述损失函数。

22.权利要求20所述的系统,其中,所述分类是包括与所述话语的最高相似度的分类。

23.一种用于对话语进行分类的系统,包括:

存储非暂态计算机可执行程序指令的计算机可读介质;以及

通信地耦合到所述计算机可读介质以执行所述非暂态计算机可执行程序指令的处理设备,其中执行所述非暂态计算机可执行程序指令将处理设备配置为执行如权利要求12-19中任一项所述的方法。

24.一种存储非暂态计算机可执行程序指令的计算机可读介质,所述计算机可执行程序指令在由处理设备执行时,使得所述处理设备执行如权利要求9-19中任一项所述的方法。

25.一种包括用于执行如权利要求9-19中任一项所述的方法的构件的装置。

使自主代理能够区分问题和请求

[0001] 相关申请的交叉引用

[0002] 本申请要求于2017年9月28日提交的美国临时申请No.62/564,868的权益,该申请通过引用整体并入本文。

技术领域

[0003] 本公开一般而言涉及语言学。更具体而言,本公开涉及使用语言学将话语分类为问题或让自主代理执行动作的请求。

[0004] 关于由联邦政府资助的研发工作所获得的发明权利的声明

[0005] 不适用

背景技术

[0006] 自主代理可以被用于解决各种用户问题。例如,自主代理从用户接收对信息的请求,诸如“what are the fees associated with a checking account (与支票账户相关的费用是多少)”。随后,用户可能会问“please open a new bank account for me. (请为我开设新的银行账户)”。

[0007] 但是,当前的解决方案不能区分与问题相关的用户话语 (utterance) 和作为对要执行的动作的请求的话语。这部分地是因为是请求可能是隐式的,例如,“I am too cold (我太冷)”,而不是显式的,例如“please turn on the heat. (请打开暖气)”。现有的解决方案可能将“I am too cold”错误地分类为与“coldness(冷)”相关的问题,并提供诸如“in San Francisco, the temperature can sometimes be cold. (在旧金山,温度有时可能会很冷)”之类的答案。

[0008] 因此,现有的解决方案可能使用户感到沮丧。继续以上示例,用户可能会感到沮丧,因为他对自主代理打开暖气的期望被忽略。误认这两种类型的话语的后果可能很严重。例如,如果代理误解了用户的问题,那么用户可以重新制定该问题,但是如果代理仍然无法识别该问题,那么用户可能会对如何继续感到困惑或者干脆完全放弃使用代理,从而导致代理没太大用。用户的沮丧反过来可能导致自主代理的低采用率。

[0009] 由此,需要用于在问题和请求之间进行区分的解决方案。

发明内容

[0010] 一般而言,本发明的系统、设备和方法与文本分类相关。在一方面,一种系统包括语言引擎(linguistic engine)和规则引擎(rule engine)。语言引擎被配置为访问包括至少一个词语(word)的话语。语言引擎还被配置为生成用于该话语的解析树。解析树包括包含词语类型的至少一个非终端节点和包含类型的至少一个终端节点。每个终端节点表示话语的词语或短语(phase)。语言引擎还被配置为将包含一个或多个词语类型的一个或多个语言模板应用于解析树。规则引擎被配置为生成从语言引擎获得的一个或多个匹配结果,并通过应用规则来确定文本的分类。规则包括:响应于确定语言模板匹配而将话语分类为

请求。规则还包括：响应于确定话语包括祈使动词(imperative verb)作为话语的第一词语而将话语分类为请求。规则还包括：响应于在话语中识别一个或多个预定义的请求关键字而将话语分类为请求。规则还包括：响应于在话语中识别一个或多个预定义的问题关键字，将话语分类为问题。该系统还基于分类被配置为向用户设备发送消息或调整外部设备的配置。

[0011] 在一方面，规则还包括：响应于未能将话语识别为问题或请求而将话语分类为未知的。

[0012] 在一方面，语言模板包括名词和第一人称(first-person)代词。

[0013] 在一方面，规则还包括：响应于确定话语在第一词语位置中包括问题前缀而将话语分类为问题。

[0014] 在一方面，一个或多个预定义的问题词语包括心理动词。

[0015] 在一方面，规则还包括：响应于确定话语在话语的最后词语位置处包括请求后缀而将话语分类为请求。

[0016] 在一方面，规则还包括：响应于确定话语包括事务动词(transaction verb)而将话语分类为请求。

[0017] 在一方面，语言引擎被配置为：在确定解析树之前，从话语中移除一个或多个停用词(stop word)。

[0018] 在一方面，一种使用分类模型将话语分类为问题或请求的方法访问文本的话语。话语包括至少一个词语。该方法生成用于该话语的解析树。解析树包括包含词语类型的至少一个非终端节点和包含类型的至少一个终端节点。每个终端节点表示话语的词语或短语。该方法通过将分类模型应用于解析树来确定话语的分类。确定话语的分类包括为解析树确定第一相似度分数，该第一相似度分数指示话语和被识别为话语的训练类别之间的第一匹配。确定话语的分类包括为解析树确定第二相似度分数，该第二相似度分数指示话语和被识别为请求的训练类别之间的第二匹配。该方法包括从分类模型接收话语的分类。该分类基于所确定的类别的相似度分数高于其它相似度分数，或者是未知的。

[0019] 在一方面，该方法访问包括训练数据对集合的训练数据集合。每个训练数据对包括文本和预期分类。训练数据集合既包括包含问题的第一训练数据对，又包括包含请求的第二训练数据对。该方法通过迭代地向分类模型提供训练数据对之一、从分类模型接收所确定的分类、通过计算所确定的分类与期望分类之间的差来计算损失函数，以及调整分类模型的内部参数以最小化损失函数，来训练分类模型。

[0020] 在一方面，分类是包括与话语的最高相似度的分类。

[0021] 以上方法可以在有形的计算机可读介质上实现和/或在计算机处理器和附接的存储器内操作。

附图说明

[0022] 图1示出了根据一方面的文本分类环境的示例。

[0023] 图2描绘了根据一方面的解析树的示例。

[0024] 图3描绘了根据一方面的在文本分类环境中使用的分类数据的示例。

[0025] 图4描绘了根据一方面的图示用于在问题和请求之间进行区分的处理的示例的流

程图。

[0026] 图5描绘了根据一方面的图示用于在问题和请求之间进行区分的规则的示例的流程图。

[0027] 图6描绘了根据一方面的用于训练机器学习模型以检测话语是问题还是请求的示例性处理的流程图。

[0028] 图7描绘了用于实现这些方面之一的分布式系统的简化图。

[0029] 图8是根据一方面的系统环境的部件的简化框图,通过该系统环境,可以将方面系统的部件所提供的服务作为云服务供应。

[0030] 图9图示了其中可以实现本发明的各方面的示例性计算机系统。

具体实施方式

[0031] 本文公开的方面通过提供改进的文本分类而提供了对计算机实现的语言学领域的技术改进。更具体而言,某些方面使用语言学来确定文本是问题还是对要执行的动作的请求。如以上所讨论的,用于自主代理的现有解决方案无法在问题和事务请求之间进行区分,从而导致代理与用户之间的交互失败。

[0032] 两个人之间的对话(conversation)是一种言语形式。除了更传统的手段,诸如电子邮件和语音对话之外,第一人与第二人之间的对话可以经由电子消息(例如,聊天)来进行。自主代理聊天机器人(ChatBot)、自主数字助理或虚拟助手是可以代替第二个人并在不同程度上模仿两个人之间的对话的“智能”机器。这种系统的一个目标是第一人不能分辨出第二人是机器(由艾伦图灵(Alan Turing)在1950年开发的图灵测试)。

[0033] 例如,用户可以通过对话交互来与自主代理进行交互。也称为对话用户界面的这种交互是最终用户和代理之间的会话(dialog),就像在两个人之间一样。它可能很简单,例如最终用户对代理说“你好”,然后代理回答“嗨”并询问用户它如何提供帮助,或者可能是银行代理中的事务交互,诸如将资金从一个账户转移到另一个账户,或在HR代理中的信息交互,诸如检查剩余假期,或询问零售代理中的FAQ,诸如如何处理退货。

[0034] 与其它方法结合的自然语言处理(NLP)和机器学习(ML)算法可以被用于对最终用户的意图进行分类。高层的意图是最终用户想要完成的目标(例如,获得账户余额、进行购买)。意图可以是客户输入到后端应执行的工作单元的映射。意图也可以是导致特定代理动作的一类话语(例如,请求)。因此,基于用户在代理中给出的短语,将它们映射到特定且离散的用例或工作单元,例如,支票余额、转移资金和跟踪支出都是代理应支持并且能够确定由最终用户以自然语言键入的自由文本条目应触发哪个工作单元的“用例”。

[0035] 在高层,通常有两种类型的请求:(1)执行某一动作的请求;以及(2)信息请求,例如问题。请求具有响应,在该响应中创建工作单元。问题具有对该问题的响应,例如,好的答案。在一些方面,答案可以采取以下形式:代理根据其广泛的(一个或多个)知识库来构造答案,或者通过搜索互联网或内部网或其它公共或私有可用数据源来匹配最佳现有答案。

[0036] 言语(discourse)分析,包括机器学习在内的人工智能以及自然语言处理在朝着通过图灵测试的长期目标方面取得了长足的进步。随着计算机越来越具有搜索和处理大量数据储存库以及对数据执行复杂分析以包括预测性分析的能力,长期目标是使代理像人一样并且将计算机组合在一起。但是,此类解决方案无法区分问题和事务请求。

[0037] 要将话语分类为问题或请求,某些方面通过解析树和模板结合关键字分析来使用语言分析。某些关键字(诸如祈使动词)可以指示话语是对动作的请求。类似地,话语是否包括特定类型的词语(诸如心理动词或特定前缀)的某些组合可以指示该句子是否是对要执行的动作的请求。某些方面用机器学习来补充语言处理,例如,以进一步改进分析或允许定制。

[0038] 话语可以包括显式制定的请求(例如,“please turn up the heat(请调高暖气)”或隐式地请求(例如,“it is cold.(很冷)”)。事务请求可以隐蔽为问题,例如,简单的问题“what is my account balance(我的账户余额是多少)”可以是选择账户并执行数据库查询以确定账户余额的事务请求,此外,用户可以请求期望的状态而不是为达到该状态而进行的显式操作。“I am too cold(我太冷)”不是指示问题,而是指示可以通过打开加热器实现的期望的状态。如果没有与“cold(冷)”相关联的可用动作,那么该话语可能被分类为与“(coldness)冷”相关联的问题。通过在语言上区分问题和事务请求,本文描述的各方面可以以域独立的方式处理这种看似歧义的短语。

[0039] 如所讨论的,在识别问题与事务请求时的错误可能成问题。如果自主代理误解了用户问题并返回另一个问题的答案,那么用户可以重新制定问题并再次询问。如果自主代理将事务请求识别为不期望的事务,那么用户将在代理发出指定不适当的参数的请求时认识到这一点。随后,用户可以取消她的请求并尝试重新制定该请求。但是,虽然可以纠正与被错误理解的问题和事务请求相关联的代理错误,但在识别问题与事务请求时的代理错误会打断对话流程,并使用户对如何继续对话感到困惑,从而使用户感到沮丧,并且可能拒绝自主代理。

[0040] 此外,区分问题和事务请求必须是领域独立的(例如,不依赖于特定主题,诸如银行、客户服务、研究等)。某些方面减轻了自主代理系统开发人员至少针对问题与事务请求用特定于领域的知识来训练机器的负担。此外,某些方面以上下文无关的方式操作。例如,自主代理可以处理与问题和请求相关的对话之间的来回切换。

[0041] 提供以下非限制性示例以介绍某些方面。执行分类应用的自主代理从用户接收指出“how do I check my balance?(如何检查我的余额?)”的消息。分类应用访问包括预定义的词汇表和模板的一组分类数据。分类应用对话语执行语言处理(包括解析和语义匹配)、对话语应用一组规则,并将话语分类为问题。进而,自主代理向用户设备提供关于如何检查银行余额的信息。

[0042] 某些定义

[0043] 如本文所使用的,“话语”是任何形式(诸如口头或书面形式)的一个或多个词语。示例话语包括“please turn on the lights(请打开灯)”和“what’s the weather?(天气如何?)”

[0044] 如本文所使用的,“解析树”或“语法树”是指表示文本的句法结构的有序、有根的树。例如。

[0045] 如本文所使用的,“词语类型”是指词语的属性。示例包括动词、名词、名词短语、介词等。

[0046] 如本文所使用的,“前缀”是指在句子中的所有其它词语之前的词语或词语的一部分。示例包括“please(请)”、“kindly(仁慈地)”或“help(帮助)”。

[0047] 如本文所使用的,“后缀”是指在句子中的所有其它词之后的词语或词语的一部分。示例包括“off(关闭)”或“on(打开)”。

[0048] 现在转到各图,图1示出了根据一方面的文本分类环境的示例。图1描绘了文本分类环境100,其包括用户设备170、自主代理101和外部设备180。用户设备170被配置为经由消息与自主代理101通信。继而,自主代理101分析用户的消息以确定用户的意图,特别是消息是指示问题还是请求。响应于用户请求,可以激活或重新配置外部设备180。外部设备的示例包括灯、车库门、音乐播放器、汽车等。

[0049] 用户设备170可以是任何移动设备,诸如移动电话、智能电话、平板电脑、笔记本电脑、智能手表等。用户设备170的功能可以例如经由应用或web应用软件来实现。用户设备170包括显示器171。显示器171的示例包括计算机屏幕、移动设备屏幕、LCD或LED背光显示器等。显示器171示出了四个消息181-184。

[0050] 自主代理101包括分类应用102、分类数据120、语言引擎130、规则引擎140、分类模型150或训练数据160中的一个或多个。分类数据120可以包括由语言引擎130和规则引擎140使用的词汇、模板和配置部件。例如,分类数据120包括指示话语是请求的前导动词(leading verb)的列表。示例包括如在“turn on the lights(打开灯)”中使用的动词“turn(打开)”和如在“open the garage door(开启车库门)”中使用的“open(开启)”。此外,分类数据120包括在话语中存在的指示用户期望某物的表达。示例包括诸如“Please do...for me(请为我……做)”之类的短语或诸如“Give me my...(把我的……给我)”之类的指示信息请求的短语。这些表达还包括其中用户可能陈述问题的不同方式,诸如“please tell me...(请告诉我……)”。关于图3解释了分类数据120中包括的数据的进一步示例。

[0051] 语言引擎130执行对话语的语言分析的一个或多个阶段。例如,语言引擎130可以检测前导动词的存在和对所有格(例如,“my house”)中对象的引用。语言引擎130包括解析器131和匹配器132。匹配器132可以将一个或多个模板应用于经解析的话语以确定话语是问题还是请求。模板的示例包括字符串(string)模板和语言模板。字符串模板检测话语中字符串的存在,而语言模板确定一个或多个特定类型的词语(例如,名词、动词、代词)的匹配的存在。

[0052] 规则引擎140应用一系列基于关键字、基于词汇和基于语言的规则。可以按特定顺序应用这些规则。在一方面,规则引擎140尝试识别话语中的请求。如果没有识别出请求,那么规则引擎140可以退回到默认决策并将话语分类为问题。但是指示问题的规则也是可能的。分类应用102接收由规则引擎140做出的决定并输出分类。

[0053] 在示例中,用户设备170与自主代理101通信以促进用户问题和请求。分类应用102从用户设备170接收消息181。消息181是用户话语,读为“Transfer funds from checking to savings.(将资金从支票转移到储蓄)”。继续该示例,分类应用102确定前置祈使动词“transfer(转移)”的存在,并确定消息181是请求。自主代理102通过向用户设备170发送消息182来提示用户“please confirm the amount(请确认金额)”。随后,用户设备170向自主代理101发送后续消息183,读为“how do I check my balance?(如何检查我的余额?)”。进而,分类应用102确定用户的意图(具体而言,对信息的期望),并发送回消息184以帮助用户检查其余额。

[0054] 此外,某些方面可以使用机器学习来自动化规则的应用。例如,某些方面使用分类

模型150来获得问题与请求的分类。可以使用分类模型150代替规则引擎140或作为规则模型140的附加。训练数据160包含训练对,训练对包括预期分类的文本(问题或请求)。分类应用102可以使用训练数据160来训练分类模型150。除了自主代理之外,本文描述的方面的实际应用还包括移动云实现方案(包含自主移动云企业)。

[0055] 图2描绘了根据一方面的解析树的示例。图2描绘了解析树200,其由解析器131从句子“Turn on the light. (打开灯)”生成。解析树200包括节点201-204。每个节点由类型指示,而类型又可以通过附加分析进一步细化。表1描述了类型的示例,但其它示例也是可能的。

[0056]	符号	描述
	S	句子
	NP	名词短语
	VP	动词短语
	V	动词
[0057]	D 或 DOBJ	限定词
	N	名词
	RP	短语助动词
表 1		

[0058] 如从图2中可以看到的,解析器131确定如由节点201所示的动词短语“turn”、节点202处的短语助动词“on”、节点203处的“the”和节点204处的“light”。它们用不同的相互关系链接,诸如“dobj”和“det”。可以使用标准解析器,诸如Stanford NLP解析器。

[0059] 分类应用102可以基于特定预期词语或类型在话语内的位置来识别一个或多个关键词或使一个或多个语言模板与话语匹配。例如,节点202指代词语“Turn”,其位于话语内的第一位置,即开始。“On”节点203位于话语内的第二位置,依此类推。节点204或者说“lights”位于最终位置,或者说在话语结束处。

[0060] 图3描绘了根据一方面的在文本分类环境中使用的分类数据的示例。分类数据301是图1中描绘的分类数据120的示例。分类应用102使用分类数据301来辅助解析、语法匹配和语言处理。分类数据301包括问题意图前缀311、心理动词312、祈使动词313、请求后缀314、请求前缀315、请求动词316、停用前缀317和模板318。其它词汇词语或由分类应用102随着时间学习的词语也是可能的。分类数据301可以布置为一个或多个阵列、列表、数据库或其它数据结构。

[0061] 分类应用102通过使用分类数据301来识别话语中的一个或多个词汇或模板匹配。但是每个匹配结果都可以由规则集合进行加权,如关于图4进一步解释的。由此,单独的特

定匹配并不必然确定特定的分类。

[0062] 问题意图前缀311包括前缀词语,前缀词语帮助分类应用102识别话语内的问题。例如,包括“I want to know(我想知道)”的话语包括意图前缀“I want(我想)”。表2中示出了意图前缀的示例。问题意图前缀311包括简写(shorthand)或基于互联网的通信可能预期的拼写错误和非正式变体。

[0063]

问题意图前缀	
can you tell	can you explain
please tell	please inform
why	how to
how much can	how do i
do i	how can
what' s the cause	whats the cause
what is the cause	what s the cause
what' s the difference	whats the difference
what is the difference	what s the difference
what' s the reason	whats the reason
what is the reason	what s the reason
how much money	show how
show me how	i need to know
i want to know	where can
表 2	

[0064] 分类应用102可以维护心理动词的数据库312。心理动词312包括与心理活动相关的动词。示例包括“know(知道)”、“think(认为)”、“recognize(认识到)”等。心理动词312可以包括每个动词以不同时态的变体(例如,know、knew、thought等)。在检测到一个或多个心理动词312的存在时,分类应用102可以将话语310分类为问题。

[0065] 在使用祈使动词313的情况下,匹配器132可以识别话语中的一个或多个祈使动词313。祈使动词313包括祈使动词、动词短语以及诸如拼写错误、标点不正确的动词或同一动词的不同时态之类的变体。在检测到一个或多个祈使动词313的存在时,分类应用102可以将话语分类为请求。表3图示了祈使动词和变体的一些示例。

[0066]

示例祈使动词短语和变体		
do not	please do not	don't
please dont	"don t"	"please don t"
i want to	"i don t want to"	"i don't want to"
"i want you do"	i dont want you to",	i don t want u to
"can you	could you	can u
could u	what is my	where are my
when are my	i need	i want
it is	is it	what is the time
what day	what year	
表 3		

[0067] 在一方面,匹配器132确定话语中的第一词语是现在时态中的动词、主动语态中的动词、并且不是情态(modal)、心理或者是词语“to be”的形式。这样的约束确保例如动词处于祈使形式,例如,“drop the temperature in the room.(降低房间中的温度)”。

[0068] 在另一方面,语言引擎130还可以检测话语是否与话语作者所拥有的或话语作者与之相关联的对象相关。示例包括“my account balance(我的账户余额)”和“my car(我的车)”。通过识别这些话语,语言引擎130可以检测针对话语中的(一个或多个)对象执行动作的意图或与对于关于(该一个或多个)对象的信息的请求,而不是表达了分享不关于该(一个或多个)特定对象的一般知识的请求的问题。

[0069] 在另一方面,分类应用102可以确定附加动词变体。动词变体包括诸如用前缀“re”、“dis”或“un”识别附加动词。示例包括“re-load”、“re-heat”、“disassemble”和“unmake”。在另一方面,分类应用102维护祈使动词前缀的列表(如在“can you turn on the light”中的“can you”)。分类应用102可以检测到此类前缀的存在,并将话语分类为请求。

[0070] 为了辅助确定话语是否是请求,分类应用102可以访问请求后缀314。请求后缀314包括描述状态(诸如对象的状态)的形容词。表4描绘了请求后缀的示例。

[0071]

请求后缀	
off	on
please	hot
cold	
表 4	

[0072] 为了辅助确定话语是否是请求,分类应用102可以访问请求前缀315。请求前缀315包括前导形容词。表5描述了事务后缀的示例。

[0073]

请求前缀	
too	more
表 5	

[0074] 请求动词316包括指示请求或事务的动词。请求动词316的示例还包括“get”、“set”、“bring”和“withdraw”。在检测到话语中的请求动词316之一后,分类应用102可以将话语分类为请求。附加地,分类数据301可以包括不指示请求或事务的一个或多个动词。示例包括“burn”或“forget”。分类应用102可以忽略话语中此类词语的存在。表6描绘了指示事务请求和不存在事务请求的动词的示例。

[0075]

指示事务请求的动词			不指示事务请求的动词
answer	get	consume	burn
approve	take	continue	bet
avoid	begin	adjust	dig
base	build	adopt	dream
block	break	check	drink
send	give	complain	feel
receive	carry	close	forgive
bill	classify	complete	forget
connect	clear	confine	hear
consider	consume	credit	hurt
continue	contribute	convert	mean
表 6			

[0076] 停用前缀317包括用户可能添加到话语开头的词语或前缀,诸如“please”。在一方面,分类应用102可以移除停用前缀,以便不干扰其它处理。表7图示了停用前缀的示例。

[0077]

停用前缀	
please	I
we	kindly
pls	
表 7	

[0078]

[0079] 分类数据301可以包括一个或多个模板318。匹配器132还可以将一个或多个语言模板应用于话语。模板318可以包括基于语法的模板或语言模板。基于语法的示例模板是话语是否包含“how to”或“if”。

[0080] 图4描绘了根据一方面的图示用于在问题和请求之间进行区分的处理的示例的流程图。

[0081] 在框401处,处理400涉及访问包括词语的话语。话语的示例包括短的短语(诸如“stop”)、较长的短语(诸如“turn off the heat (关闭暖气)”或“how do I check my portfolio? (我怎么检查我的文件夹?)”)。分类应用102可以访问来自在计算设备101上执行的处理或应用或来自另一个计算设备(诸如用户设备)的话语。

[0082] 在一方面,分类应用102对话语101进行预处理以移除停用前缀317。移除终止前缀317可以防止错误,包括忽略话语中处于第二位置或后续位置但指示特定分类的词语。

[0083] 在框402处,处理400涉及生成用于话语的解析树。分类应用102使用解析器131以与关于图2所描述的基本相似的方式来生成解析树。

[0084] 在框403处,处理400涉及基于关键字或语言分析来评估一个或多个规则。分类应用102可以使用解析树或话语(即,文本)作为任何规则的输入。

[0085] 图5描绘了根据一方面的图示用于在问题和请求之间进行区分的规则的示例的流程图。分类应用102,具体是规则引擎140,可以单独地、组合地和以任何顺序执行框501-505中的一个或多个。不同的优先顺序是可能的。例如,如果规则引擎140确定框501已经成功执行,那么规则引擎140可以输出分类“请求”,并返回到框404。在另一个示例中,如果规则引擎140执行框502,但是没有识别任何请求关键字,那么规则引擎140可以继续到框502-505之一。

[0086] 在框501处,处理500涉及识别一个或多个预定义的请求关键字。预定义的请求关键字可以包括请求后缀314、请求前缀315和请求动词316。这些关键字中的一个或多个关键字的存在指示请求。表8描绘了被识别为事务的句子,并图示了在每种情况下执行的分析。

[0087]	句子	分析
	Turn the light on	请求后缀 “on”
	Put wiper rate on high	请求后缀 “high”
	set the security system to off	请求后缀 “off”
	too loud, quieter please	请求前缀 “too”
表 8		

[0088] 更具体而言,每个类型的请求关键字可以具有该关键字预期的相关联位置。例如,分类应用102在话语的第一词语位置中搜索请求前缀、在话语中的最后一个词语位置中搜索请求后缀,并在话语中的任何位置处搜索请求动词。

[0089] 在框502处,处理500涉及确定解析树的第一终端节点包括祈使动词。匹配器132从解析器131接收表示话语的语法解析树,并从祈使动词313中识别祈使动词。如果识别出话

语的前导祈使动词或第一词语位置中的动词,那么分类应用102可以输出分类“请求”,并且处理500和400可以终止。表9描绘了基于祈使动词的存在而被识别为请求的话语的示例。

[0090]	句子	祈使动词
	Open iWish-a Flexible Recurring Deposit	open
	Cancel a Fixed Deposit using ICICI Bank VISA Debit Card	cancel
	Help me to Login and raise a Service Request	help
	Transfer funds from checking to saving	transfer
	Move funds from saving to mortgage	move
	close the garage door	Close
	do western union	do
表 9		

[0091] 在一方面,分类应用102可以权衡不同的因素。例如,在分类应用102检测到指示请求的前导动词的存在的情况下,话语中“how”的存在可以否定前导动词的存在并指示问题。在这种情况下,分类应用102将话语分类为问题。

[0092] 此外,分类应用102可以附加地检测到第一人称代词的存在,诸如“me”或“my”。更具体而言,在前导动词是通常指示问题的心理动词的情况下,“me”或“my”的存在仍然可以指示请求。因此,在心理动词与“me”或“my”结合的情况下,分类应用102将话语分类为请求。表10描绘了用这些规则识别出的话语的示例。

[0093]	句子	分析
	give me cheque deposited in Bank Account but not credited	[前导祈使动词 + me] 前导动词识别为“give”,并且识别“me”的存在。被分类为请求。
	Fund my investment account from checking	前导动词识别为“fund”,并且识别“my”而没有“how”。被分类为请求。
[0094]	Wire money from my checking to investment	前导动词识别为“fund”,并且识别“my”而没有“how”。被分类为请求。
	Thinking about checking accounts.	前导心理动词。被分类为问题。
	Thinking about my checking account.	前导心理动词“thinking”与“my”的组合。被分类为请求。
表 10		

[0095] 如果未找到前导祈使动词匹配,那么处理500可以继续进行到框501或503-505中的一个或多个,以进行进一步评估。

[0096] 在框503处,处理500涉及将一个或多个语言模板应用于解析树并确定语言模板匹配。语言模板可以包括一个或多个词语类型(例如,动词)。更具体而言,匹配器132通过确定解析树中存在一个或多个词语类型来确定匹配。模板的示例是代词后接名词(由PR+NN表示)的存在。该模板的匹配可以指示请求。例如,“give me my balance”或“get me my balance”,其中代词是“my”并且名词是“balance”。相反,由于没有代词,所以匹配器132不将话语“tell me how to check an account balance”分类为事务。

[0097] 在框504处,处理500涉及识别一个或多个预定义的问题关键字。问题关键字的示例包括问题意图前缀311和心理动词312。这些关键字中的一个或多个关键字的存在指示话语是问题。表11描绘了其中分类应用102识别出一个或多个问题关键字的句子的示例。

句子	分析
I am anxious about spending my money	[心理动词]
I am worried about my spending	[心理动词]
I am concerned about how much I used	[心理动词]
I am interested how much money I lost on stock	[心理动词]
How can my saving account be funded	[How+my]
If I do not have my Internet Banking User ID and Password, how can I login?	[if 和 “how can I” - 前缀]
表 11	

[0100] 在一些情况下,如果用于其它分类的规则未被应用,那么分类应用102可以默认为特定分类。例如,如果在话语“domestic wire transfer”中没有识别出请求,那么分类应用102将话语识别为问题。表12列出了附加情况。

句子	分析
Domestic wire transfer	[没有触发事务规则, 因此是问题]
order replacement / renewal card not received	[没有触发事务规则, 因此是问题]
表 12	

[0102] 在框503处,处理500涉及未能将话语识别为问题或请求。如果没有规则成功地将话语识别为问题或请求,那么自主代理101可以要求用户进一步澄清。

[0103] 返回图4,处理400涉及输出分类。自主代理可以接收分类并采取相应的动作。基于识别出的分类,自主代理101可以采取一些动作,诸如与用户设备170的进一步交互或者重

新配置、激活或停用外部设备180。

[0104] 如所讨论的,某些方面可以使用机器学习来对文本进行分类。示例分类模型是分类模型150,其可以使用不同的模型,诸如分类器、树核(Tree Kernels)或支持向量机或SVM TK。训练数据可以从各种来源获得。训练数据可以是特定于应用的。例如,使用某些方面的金融机构可能希望提供可接受的银行事务的列表(例如,检查余额、转移资金),使得用适当的内容训练分类应用102。

[0105] 图6描绘了根据一方面的图示用于训练分类模型以确定用于索引的信息文本的处理的示例的流程图。如处理600中进一步描述的,分类模型150可以被训练以区分问题和请求。训练数据160可以包括两个训练集,诸如具有被识别为请求的文本的训练集和具有被识别为问题的文本的第二训练集。训练数据160可以包括文本和/或相关联的解析树。

[0106] 在框601处,处理600涉及访问包括训练对集合的训练数据集。每个训练数据对包括文本和预期分类(例如,问题或请求)。训练数据集可以包括包含问题的第一训练数据对和包含请求的第二训练数据对两者。

[0107] 在框602处,处理600涉及将训练数据对之一提供给分类模型。相应地,分类模型接收文本主体和预期分类。

[0108] 在框603处,处理600涉及从分类模型接收所确定的分类。

[0109] 在框604处,处理600涉及通过计算所确定的分类与预期分类之间的差来计算损失函数。不同的损失函数是可能的,诸如均方误差、似然损失、对数(或交叉熵)损失等。

[0110] 在框605处,处理600涉及调整分类模型的内部参数以最小化损失函数。通过这种方式,分类模型利用每次训练迭代来学习以提高其预测的准确性。

[0111] 在框606处,处理600涉及使用经训练的分类模型。例如,经训练的分类模型可以结合或代替各种分类规则在处理400或500中使用。

[0112] 例如,要使用经训练的分类模型150,分类应用102可以访问文本的话语并为该话语生成解析树。分类应用102通过将分类模型150应用于解析树来确定话语的分类。

[0113] 进而,分类应用102使用几种方法之一来确定分类。例如,分类模型150确定第一相似度分数以及第二相似度分数,第一相似度分数指示话语和被识别为问题的训练类别之间的第一匹配,第二相似度分数指示话语和被识别为请求的训练类别之间的第二匹配。

[0114] 分类模型150基于第一相似度分数和第二相似度分数输出分类。例如,如果第一相似度分数高于第二相似度分数,那么分类模型150输出分类“问题”。如果第二相似度分数高于第一相似度分数,那么分类模型150输出分类“请求”。在一些情况下,例如,如果发生错误,那么分类模型150可以输出分类“未知”。

[0115] 定制分类标准

[0116] 自主代理的开发人员或供应商可以通过提供包括两个类别的样本的训练集来覆盖默认训练集或规则(即,问题与事务请求)。分类模型150的重新训练可以在使用之前或在运行时进行。例如,在接收到新话语时,分类应用102可以执行以下处理:

[0117] (1) 创建当前话语的即时索引;

[0118] (2) 遍历两个类别的所有样本、构建查询并针对即时索引进行搜索;

[0119] (3) 收集提供具有其类别的非空搜索结果的查询集合,并按这些类别对该集合进行聚合;以及

[0120] (4) 在验证某个类别由聚合结果高度表示且另一类别具有明显较低表示之后,选择高度表示的类别作为识别结果。

[0121] Lucene默认的TF*IDF模型将确保训练集元素在最重要的关键字方面最接近(从频率角度来看(Tan 2005, Salton和Yang1973))。(Trstenjak等人3014)提出了使用带有TF*IDF方法的最近邻(KNN)算法进行文本分类的可能性。这种方法使得能够根据各种参数进行分类、测量和分析结果。框架的评估集中在分类的速度和质量上,并且测试结果显示了TF*IDF-KNN算法的积极和负面特性。评估是对在线环境中的几类文档执行的,并且显示了稳定且可靠的性能。测试揭示了分类的质量并确定了哪些因素对分类的性能有影响。

[0122] 示例软件代码

[0123] 下面示出了实现分类应用102的功能的示例代码。

```
// returns null if meaningless utterance, or unable to recognize
```

```
// returns true if TRANSACTIONAL, false if QUESTION.
```

```
// explanationMap stores <decisions, with explanations>
```

```
public Boolean isARequestUtteranceKeywordBased(String utteranceOrig, Map<String,
Object> explanationMap) {
```

```
    String utterance = utteranceOrig.toLowerCase();
```

```
    Boolean matchedWithTrainingSet =
```

```
    lingRecognizer.matchWithTrainingSet(utteranceOrig);
```

```
    if (matchedWithTrainingSet != null) {
```

```
[0124]         explanationMap.put("ling_match_recognizer", true);
```

```
        explanationMap.putAll(lingRecognizer.getExplanationMap());
```

```
        return matchedWithTrainingSet;
```

```
    }
```

```
    List<String> tokens = TextProcessor.fastTokenize(utterance, false);
```

```
    if (tokens.size() < 2) {
```

```
        explanationMap.put("analysis is rejected", "single token utterance");
```

```
        return null;
```

```
    }
```

```
    // remove stop - pre - word
```

```
    if (tokens.get(0).equals("please") || tokens.get(0).equals("i") ||
```

```
tokens.get(0).equals("we"))
```

```
        tokens.remove(0);
    // remove stop-phrase
    try {
        if (tokens.get(0).equals("this") && (tokens.size() > 2 &&
tokens.get(1).equals("is")) || (tokens.size() > 2 && tokens.get(2).equals("a"))) {
            tokens.remove(0);
            tokens.remove(1);
            tokens.remove(2);
        }
    } catch (Exception e) {

    }

    // check verb + verb prefixes lookup, in case parsing will not find verbs properly
    if (requestVerbsCached.contains(tokens.get(0))) {
        [0125] explanationMap.put("check verb + verb prefixes lookup, in case parsing will
not find verbs properly. Found leading verb: ", tokens.get(0));
        explanationMap.put("reduced sentence for analysis", tokens.toString());

        return true;
    }
    // check if the first word is an (imperative) verb
    if (lingRecognizer.isARequestUtteranceKeywordBased(utteranceOrig)) {
        explanationMap.put("ling_recognizer", true);
        explanationMap.put("why transactional", "leading verb");
        explanationMap.putAll(lingRecognizer.getExplanationMap());
        return true;
    }

    // prefix + verb for request
    for (String prefix : requestVerbsPrefix) {
        if (utterance.startsWith(prefix)) {
```

```
String utteranceRemainder = utterance.replace(prefix, "");
List<String> tokensRemainder =
TextProcessor.fastTokenize(utteranceRemainder, false);
if (tokensRemainder.isEmpty()) {
    explanationMap.put("why transactional", "leading verb");
    explanationMap.put("all non-verbs are removed", "remained
empty");

    return true;
}
if (requestVerbsCached.contains(tokensRemainder.get(0))) {
    explanationMap.put("why transactional", "leading verb");
    explanationMap.put("leading verb", tokensRemainder.get(0));
    return true;
}
}
```

[0126]

```
// prefix for non-request
for (String prefix : faqVerbsPrefix) {
    if (utterance.startsWith(prefix)) {
        explanationMap.put("whyQnA", "QnA prefix");
        explanationMap.put("QnA prefix", prefix);
        return false;
    }
}

// prefix + verb for request
for (String prefix : requestVerbsPrefix) {
    if (utterance.startsWith(prefix)) {
        explanationMap.put("why transactional", "transactional prefix");
        explanationMap.put("transactional prefix", prefix);
        return true;
    }
}
```

```
}
```

```
for (String suffix : this.requestWordSuffix) {  
    if (utterance.endsWith(suffix)) {  
        explanationMap.put("why transactional", "transactional suffix");  
        explanationMap.put("transactional suffix", suffix);  
        return true;  
    }  
}
```

```
explanationMap.put("neither QnA nor Transactional rule fired", "");  
return null;  
}
```

[0127] //This is used for classifying an utterance by machine learning
// returns true if TRANSACTIONAL, false if QUESTION.
// explanationMap stores <decisions, with explanations>

```
public Boolean matchWithTrainingSet(String utterance) {  
  
    int count = 0, bestScoreIndex = -1;  
    double bestScore = 0;  
    Boolean bestDecision = null;  
    for (String[] line : nearestNeighborTrainingSet) {  
        List<List<ParseTreeChunk>> res = matcher.assessRelevanceCache(utterance,  
line[0]);  
        double syntacticScore =  
parseTreeChunkListScorer.getParseTreeChunkListScoreAggregPhraseType(res);  
        if (syntacticScore > bestScore && syntacticScore > MIN_SCORE_TO_MATCH) {  
            bestScore = syntacticScore;  
            try {
```

```
        bestDecision = Boolean.parseBoolean(line[1]);
    } catch (Exception e) {
        e.printStackTrace();
    }
    bestScoreIndex = count;
}
count++;
[0128]
}
    explanationMap.put("found decision via matching phrase", bestDecision);
    if (!nearestNeighborTrainingSet.isEmpty())
        explanationMap.put("found matching phrase",
            nearestNeighborTrainingSet.get(bestScoreIndex)[0]);
    return bestDecision;
}
```

[0129] 图7绘出了用于实现方面之一的分布式系统700的简化图。在所示方面中,分布式系统700包括一个或多个客户端计算设备702、704、706和708,其被配置为通过一个或多个网络710执行和操作客户端应用,诸如web浏览器、专有客户端(例如,Oracle Forms)等。服务器712可以经由网络710与远程客户端计算设备702、704、706和708通信地耦合。

[0130] 在各种方面中,服务器712可以适于运行由系统的一个或多个部件提供的一个或多个服务或软件应用。服务或软件应用可以包括非虚拟和虚拟环境。虚拟环境可以包括用于虚拟事件、展览、模拟器、教室、购物交易场所和企业的环境,无论是二维或者三维(3D)表示形式、基于页面的逻辑环境还是其他形式。在一些方面中,这些服务可以作为基于web的服务或云服务被提供,或者在软件即服务(SaaS)模型下被提供给客户端计算设备702、704、706和/或708的用户。操作客户端计算设备702、704、706和/或708的用户又可以利用一个或多个客户端应用来与服务器712交互以利用由这些部件提供的服务。

[0131] 在图中绘出的配置中,分布式系统700的软件部件718、720和722被示出为在服务器712上实现。在其它方面中,分布式系统700的一个或多个部件和/或由这些部件提供的服务也可以由客户端计算设备702、704、706和/或708中的一个或多个来实现。然后,操作客户端计算设备的用户可以利用一个或多个客户端应用来使用由这些部件提供的服务。这些部件可以用硬件、固件、软件或其组合来实现。应该认识到的是,各种不同的系统配置是可能的,其可能与分布式系统700不同。图中所示的方面因此是用于实现方面系统的分布式系统的一个示例,而不是要进行限制。

[0132] 客户端计算设备702、704、706和/或708可以是便携式手持设备(例如,iPhone®、蜂窝电话、iPad®、计算平板电脑、个人数字助理(PDA))或可穿戴设备(例如,Google Glass®头戴式显示器),运行诸如Microsoft Windows Mobile®和/或各种移

动操作系统(诸如iOS、Windows Phone、Android、BlackBerry 10、Palm OS等)的软件,并且启用互联网、电子邮件、短消息服务(SMS)、**Blackberry®**或其它通信协议。客户端计算设备可以是通用个人计算机,作为示例,包括运行各种版本的Microsoft **Windows®**、Apple **Macintosh®**和/或Linux操作系统的个人计算机和/或膝上型计算机。客户端计算设备可以是运行任何各种可商业获得的**UNIX®**或类UNIX操作系统(包括但不限于各种GNU/Linux操作系统,诸如例如Google Chrome OS)的工作站计算机。替代地或附加地,客户端计算设备702、704、706和708可以是能够通过(一个或多个)网络710通信的任何其它电子设备,诸如瘦客户端计算机、启用互联网的游戏系统(例如,具有或不具有**Kinect®**手势输入设备的微软Xbox游戏控制台)和/或个人消息传送设备。

[0133] 虽然示例性分布式系统700被示出为具有四个客户端计算设备,但是可以支持任何数量的客户端计算设备。其它设备(诸如具有传感器的设备等)可以与服务器712交互。

[0134] 分布式系统700中的(一个或多个)网络710可以是本领域技术人员熟悉的、可以利用任何各种可商业获得的协议支持数据通信的任何类型的网络,其中协议包括但不限于TCP/IP(传输控制协议/网际协议)、SNA(系统网络体系架构)、IPX(互联网报文交换)、AppleTalk,等等。仅作为示例,(一个或多个)网络710可以是局域网(LAN),诸如基于以太网、令牌环等的LAN。(一个或多个)网络710可以是广域网和互联网。它可以包括虚拟网络,包括但不限于虚拟专用网(VPN)、内联网、外联网、公共交换电话网(PSTN)、红外网络、无线网络(例如,依据电子电气学会(IEEE)802.7协议套件、**Bluetooth®**和/或任何其它无线协议当中任意一种操作的网络);和/或这些和/或其它网络的任意组合。

[0135] 服务器712可以由一个或多个通用计算机、专用服务器计算机(作为示例,包括PC(个人计算机)服务器、**UNIX®**服务器、中档服务器、大型计算机、机架安装的服务器等)、服务器场、服务器集群或任何其它适当的布置和/或组合组成。服务器712可以包括运行虚拟操作系统或涉及虚拟化的其他计算架构的一个或多个虚拟机。可以虚拟化一个或多个逻辑存储设备的灵活的池,以维护服务器的虚拟存储设备。服务器712可以使用软件定义的网络来控制虚拟网络。在各种方面中,服务器712可以适于运行在前述公开中所描述的一个或多个服务或软件应用。例如,服务器712可以与用于执行以上根据本公开的方面描述的处理的服务器对应。

[0136] 服务器712可以运行包括以上讨论的操作系统当中任意一种的操作系统,以及任何可商业获得的服务器操作系统。服务器712还可以运行任何各种附加的服务器应用和/或中间层应用,包括HTTP(超文本传输协议)服务器、FTP(文件传输协议)服务器、CGI(公共网关接口)服务器、**JAVA®**服务器、数据库服务器,等等。示例性数据库服务器包括但不限于从Oracle、Microsoft、Sybase、IBM(国际商业机器)等可商业获得的那些数据库服务器。

[0137] 在一些实现中,服务器712可以包括一个或多个应用,以分析和整合从客户端计算设备802、804、806和808的用户接收到的数据馈送和/或事件更新。作为示例,数据馈送和/或事件更新可以包括,但不限于,**Twitter®**馈送、**Facebook®**更新或者从一个或多个第三方信息源接收到的实时更新和连续数据流,其可以包括与传感器数据应用、金融报价机、网络性能测量工具(例如,网络监视和流量管理应用)、点击流分析工具、汽车交通监视

等相关的实时事件。服务器712还可以包括一个或多个应用,以经由客户端计算设备702、704、706和708的一个或多个显示设备显示数据馈送和/或实时事件。

[0138] 分布式系统700还可以包括一个或多个数据库714和716。数据库714和716可以驻留在各种位置中。作为示例,数据库714和716中的一个或多个可以驻留在服务器712本地的(和/或驻留在服务器712中的)非瞬态存储介质上。替代地,数据库714和716可以远离服务器712,并且经由基于网络的连接或专用的连接与服务器712通信。在一组方面中,数据库714和716可以驻留在本领域技术人员熟悉的存储区域网络(SAN)中。类似地,用于执行服务器712所具有的功能的任何必要的文件都可以适当地本地存储在服务器712上和/或远程存储。在一组方面中,数据库714和716可以包括适于响应于SQL格式的命令而存储、更新和检索数据的关系数据库,诸如由Oracle提供的数据库。

[0139] 图8是根据本公开的方面的系统环境800的一个或多个部件的简化框图,通过该系统环境800,由方面系统的一个或多个部件提供的服务可以作为云服务提供。在所示方面中,系统环境800包括可以由用户使用以与提供云服务的云基础设施系统802交互的一个或多个客户端计算设备804、806和808。客户端计算设备可以被配置为操作客户端应用,诸如web浏览器、专有客户端应用(例如,Oracle Forms)或某种其它应用,这些应用可以由客户端计算设备的用户用来与云基础设施系统802交互以使用由云基础设施系统802提供的服务。

[0140] 应该认识到的是,图中描绘的云基础设施系统802可以具有除了所描绘的那些之外的其它部件。另外,图中所示的方面仅是可以结合本发明的方面的云基础设施系统的一个示例。在一些其它方面中,云基础设施系统802可以具有比图中所示更多或更少的部件、可以组合两个或更多个部件、或者可以具有不同的部件配置或布置。

[0141] 客户端计算设备804、806和808可以是与上面针对702、704、706和708所描述的设备类似的设备。

[0142] 虽然示例性系统环境800被示出具有三个客户端计算设备,但是任何数量的客户端计算设备可以被支持。诸如具有传感器的设备等的其它设备可以与云基础设施系统802交互。

[0143] (一个或多个)网络810可以促进客户端804、806和808与云基础设施系统802之间的数据通信和交换。每个网络可以是本领域技术人员所熟悉的可以使用各种商业上可获得的协议(包括上面针对(一个或多个)网络710所描述的那些协议)中的任何一种支持数据通信的任何类型的网络。

[0144] 云基础设施系统802可以包括一个或多个计算机和/或服务器,其可以包括上面针对服务器812所描述的那些计算机和/或服务器。

[0145] 在某些方面中,由云基础设施系统提供的服务可以包括按需对云基础设施系统的用户可用的许多服务,诸如在线数据存储和备份解决方案、基于Web的电子邮件服务、被托管的办公室(office)套件和文档协作服务、数据库处理、受管理的技术支持服务等。由云基础设施系统提供的服务可以动态扩展以满足云基础设施系统的用户的需要。由云基础设施系统提供的服务的具体实例化在本文中被称为“服务实例”。一般而言,从云服务提供商的系统经由通信网络(诸如互联网)对用户可用的任何服务被称为“云服务”。通常,在公共云环境中,构成云服务提供商的系统的服务器和系统与用户自己的本地服务器和系统不同。

例如,云服务提供商的系统可以托管应用,并且用户可以经由诸如互联网的通信网络按需订购和使用应用。

[0146] 在一些示例中,计算机网络云基础设施中的服务可以包括对存储装置、被托管的数据库、被托管的Web服务器、软件应用或由云供应商向用户提供的其它服务的受保护的计算机网络访问,或者如本领域中另外已知的那样。例如,服务可以包括通过互联网对云上的远程存储装置进行密码保护的访问。作为另一个示例,服务可以包括基于Web服务的被托管的关系数据库和脚本语言中间件引擎,以供联网的开发人员私有使用。作为另一个示例,服务可以包括对在云供应商的网站上托管的电子邮件软件应用的访问。

[0147] 在某些方面中,云基础设施系统802可以包括以自助服务、基于订阅、弹性可扩展、可靠、高度可用和安全的方式递送给客户的应用、中间件和数据库服务产品的套件。这种云基础设施系统的示例是由本受让人提供的Oracle公共云。

[0148] 大量数据(有时称为大数据)可以由基础设施系统在许多级别和不同规模上托管和/或操纵。这样的数据可能包含庞大而复杂的数据集,以致于很难使用典型的数据库管理工具或传统的数据处理应用进行处理。例如,使用个人计算机或其基于机架的对应物可能难以存储、检索和处理兆字节级的数据。使用最新的关系数据库管理系统以及桌面统计数据可视化程序包,可能难以处理这种大小的数据。他们可能需要运行数千台服务器计算机的超出常用软件工具的结构的大规模并行处理软件,才能在可容忍的经过时间内捕获、整理、管理和处理数据。

[0149] 分析人员和研究人员可以存储和操纵非常大的数据集,以可视化大量数据,检测趋势和/或以其他方式与数据进行交互。并行链接的数十个、数百个或数千个处理器可以对此类数据进行操作,以呈现该数据或模拟对该数据或其表示的外力。这些数据集可能涉及结构化数据(例如,在数据库中组织的数据或根据结构化模型进行组织的数据)和/或非结构化数据(例如,电子邮件、图像、数据块(二进制大对象)、网页、复杂事件处理)。通过利用一个方面的能力来相对快速地将更多(或更少)计算资源集中在一个目标上,可以根据企业、政府机构、研究组织、私人、志趣相投的个人或组织或其他实体的需求,更好地利用云基础设施系统来对大型数据集执行任务。

[0150] 在各种方面中,云基础设施系统802可以适于自动供应、管理和跟踪客户对由云基础设施系统802供给的服务的订阅。云基础设施系统802可以经由不同的部署模型来提供云服务。例如,可以依据公共云模型提供服务,其中云基础设施系统802被销售云服务的组织拥有(例如,被Oracle拥有),并且服务对一般公众或不同行业的企业可用。作为另一个示例,可以依据私有云模型来提供服务,其中云基础设施系统802仅针对单个组织操作,并且可以为该组织内的一个或多个实体提供服务。还可以依据社区云模型来提供云服务,其中云基础设施系统802和由云基础设施系统802提供的服务由相关社区中的若干组织共享。云服务还可以依据混合云模型被提供,该混合云模型是两个或更多个不同模型的组合。

[0151] 在一些方面中,由云基础设施系统802提供的服务可以包括在软件即服务(SaaS)类别、平台即服务(PaaS)类别、基础设施即服务(IaaS)类别或包括混合服务的其它服务类别下提供的一个或多个服务。客户经由订阅订单可以订购由云基础设施系统802提供的一个或多个服务。云基础设施系统802然后执行处理以提供客户的订阅订单中的服务。

[0152] 在一些方面中,由云基础设施系统802提供的服务可以包括但不限于应用服务、平

台服务和基础设施服务。在一些示例中,应用服务可以由云基础设施系统经由SaaS平台提供。SaaS平台可以被配置为提供落入SaaS类别的云服务。例如,SaaS平台可以提供在集成开发和部署平台上构建和递送按需应用套件的能力。SaaS平台可以管理和控制用于提供SaaS服务的底层软件和基础设施。通过利用由SaaS平台提供的服务,客户可以利用在云基础设施系统上执行的应用。客户可以获取应用服务,而无需客户购买单独的许可和支持。可以提供各种不同的SaaS服务。示例包括但不限于为大型组织提供销售绩效管理、企业集成和业务灵活性的解决方案的服务。

[0153] 在一些方面中,平台服务可以由云基础设施系统经由PaaS平台提供。PaaS平台可以被配置为提供落入PaaS类别的云服务。平台服务的示例可以包括但不限于使组织(诸如Oracle)能够在共享的公共体系架构上整合现有应用以及充分利用平台提供的共享服务来构建新应用的能力的服务。PaaS平台可以管理和控制用于提供PaaS服务的底层软件和基础设施。客户可以获取由云基础架构系统提供的PaaS服务,而无需客户购买单独的许可和支持。平台服务的示例包括但不限于Oracle Java云服务(JCS)、Oracle数据库云服务(DBCS)等。

[0154] 通过利用由PaaS平台提供的服务,客户可以采用由云基础设施系统支持的编程语言和工具,并且还控制所部署的服务。在一些方面中,由云基础设施系统提供的平台服务可以包括数据库云服务、中间件云服务(例如,Oracle融合中间件服务)和Java云服务。在一个方面中,数据库云服务可以支持共享服务部署模型,该模型使得组织能够汇集数据库资源并且以数据库云的形式向客户供应数据库即服务。在云基础设施系统中,中间件云服务可以为客户提供开发和部署各种业务应用的平台,并且Java云服务可以为客户提供部署Java应用的平台。

[0155] 各种不同的基础设施服务可以由云基础设施系统中的IaaS平台提供。基础设施服务促进底层计算资源(诸如存储装置、网络和其它基础计算资源)的管理和控制,以供客户利用由SaaS平台和PaaS平台提供的服务。

[0156] 在某些方面中,云基础设施系统802还可以包括基础设施资源830,用于向云基础设施系统的客户提供用于提供各种服务的资源。在一个方面中,基础设施资源830可以包括预先集成和优化的硬件(诸如服务器、存储装置和联网资源)的组合,以执行由PaaS平台和SaaS平台提供的服务。

[0157] 在一些方面中,云基础设施系统802中的资源可以由多个用户共享并且根据需要动态重新分配。此外,可以将资源分配给在不同时区的用户。例如,云基础设施系统830可以使在第一时间区中的第一组用户能够在指定的小时数内利用云基础设施系统的资源,并且然后使相同资源能够被重新分配给位于不同时间区的另一组用户,从而使资源的利用率最大化。

[0158] 在某些方面中,可以提供由云基础设施系统802的不同部件或模块以及由云基础设施系统802提供的服务共享的多个内部共享服务832。这些内部共享服务可以包括但不限于:安全和身份服务、集成服务、企业储存库服务、企业管理器服务、病毒扫描和白名单服务、高可用性、备份和恢复服务、启用云支持的服务、电子邮件服务、通知服务、文件传输服务等。

[0159] 在某些方面中,云基础设施系统802可以提供云基础设施系统中的云服务(例如,

SaaS、PaaS和IaaS服务)的综合管理。在一个方面中,云管理功能可以包括用于供应、管理和跟踪由云基础设施系统802接收到的客户订阅等的功能。

[0160] 在一个方面中,如图中所绘出的,云管理功能可以由一个或多个模块提供,诸如订单管理模块820、订单编排模块822、订单供应模块824、订单管理和监视模块826,以及身份管理模块828。这些模块可以包括一个或多个计算机和/或服务器或者使用一个或多个计算机和/或服务器来提供,这些计算机和/或服务器可以是通用计算机、专用服务器计算机、服务器场、服务器集群或任何其它适当的布置和/或组合。

[0161] 在示例性操作834中,使用客户端设备(诸如客户端设备804、806或808)的客户可以通过请求由云基础设施系统802提供的一个或多个服务并且下订阅由云基础设施系统802供应的一个或多个服务来的订单来与云基础设施系统802交互。在某些方面中,客户可以访问云用户界面(UI)(云UI 88、云UI 814和/或云UI 816)并经由这些UI下订阅订单。云基础设施系统802响应于客户下订单而接收到的订单信息可以包括识别客户以及客户想要订阅的云基础设施系统802供应的一个或多个服务的信息。

[0162] 在客户下订单之后,经由云UI 88、814和/或816接收订单信息。

[0163] 在操作836处,订单存储在订单数据库88中。订单数据库88可以由云基础设施系统88操作和与其它系统元件一起操作的若干数据库之一。

[0164] 在操作838处,订单信息被转发到订单管理模块820。在一些情况下,订单管理模块820可以被配置为执行与订单相关的计费 and 记账功能,诸如验证订单、以及在验证后预订订单。

[0165] 在操作840处,将关于订单的信息传送到订单编排模块822。订单编排模块822可以利用订单信息为客户下的订单编排服务和资源的供应。在一些情况下,订单编排模块822可以使用订单供应模块824的服务来编排资源的供应以支持所订阅的服务。

[0166] 在某些方面中,订单编排模块822使得能够管理与每个订单相关联的业务过程并应用业务逻辑来确定订单是否应该进行到供应。在操作842处,在接收到新订阅的订单时,订单编排模块822向订单供应模块824发送分配资源并配置履行订阅订单所需的那些资源的请求。订单供应模块824使得能够为客户订购的服务分配资源。订单供应模块824提供在由云基础设施系统800提供的云服务 and 用于供应于提供所请求的服务的资源的物理实现层之间的抽象层。因此,订单编排模块822可以与实现细节(诸如服务和资源是否实际上即时供应或预先供应并仅在请求后才分配/指派)隔离。

[0167] 在操作844处,一旦供应了服务和资源,就可以通过云基础设施系统802的订单供应模块824向客户端设备804、806和/或808上的客户发送所提供的服务的通知。

[0168] 在操作846处,订单管理和监视模块826可以管理和跟踪客户的订阅订单。在一些情况下,订单管理和监视模块826可以被配置为收集订阅订单中的服务的使用统计信息,诸如,所使用的存储量、传输的数据量、用户的数量,以及系统运行时间量和系统停机时间量。

[0169] 在某些方面中,云基础设施系统800可以包括身份管理模块828。身份管理模块828可以被配置为提供身份服务,诸如云基础设施系统800中的访问管理和授权服务。在一些方面中,身份管理模块828可以控制关于希望利用由云基础设施系统802提供的服务的客户的信息。这样的信息可以包括认证这些客户的身份的信息以及描述这些客户被授权相对于各种系统资源(例如,文件、目录、应用、通信端口、存储器段等)执行哪些动作的信息。身份管

理模块828还可以包括对关于每个客户的描述性信息以及关于如何和由谁来访问和修改这些描述性信息的管理。

[0170] 图9示出了其中可以实现本发明的各种方面的示例性计算机系统900。系统900可以用于实现上述任何计算机系统。如图所示,计算机系统900包括经由总线子系统902与多个外围子系统通信的处理单元904。这些外围子系统可以包括处理加速单元906、I/O子系统908、存储子系统918和通信子系统924。存储子系统918包括有形计算机可读存储介质922和系统存储器910。

[0171] 总线子系统902提供用于让计算机系统900的各种部件和子系统按意图彼此通信的机制。虽然总线子系统902被示意性地示出为单条总线,但是总线子系统的替代方面可以利用多条总线。总线子系统902可以是若干种类型的总线结构中的任何一种,包括存储器总线或存储器控制器、外围总线、以及使用任何各种总线体系架构的局部总线。例如,这种体系架构可以包括工业标准体系架构 (ISA) 总线、微通道体系架构 (MCA) 总线、增强型ISA (EISA) 总线、视频电子标准协会 (VESA) 局部总线和外围部件互连 (PCI) 总线,其可以被实现为按IEEE P986.1标准制造的Mezzanine总线。

[0172] 可以被实现为一个或多个集成电路(例如,常规微处理器或微控制器)的处理单元904控制计算机系统900的操作。一个或多个处理器可以被包括在处理单元904中。这些处理器可以包括单核或多核处理器。在某些方面中,处理单元904可以被实现为一个或多个独立的处理单元932和/或934,其中在每个处理单元中包括单核或多核处理器。在其它方面中,处理单元904也可以被实现为通过将两个双核处理器集成到单个芯片中形成的四核处理单元。

[0173] 在各种方面中,处理单元904可以响应于程序代码执行各种程序并且可以维护多个并发执行的程序或进程。在任何给定的时间,要被执行的程序代码中的一些或全部代码可以驻留在(一个或多个)处理器904中和/或存储子系统918中。通过适当的编程,(一个或多个)处理器904可以提供上述各种功能。计算机系统900可以附加地包括处理加速单元906,其可以包括数字信号处理器(DSP)、专用处理器,等等。

[0174] I/O子系统908可以包括用户接口输入设备和用户接口输出设备。用户接口输入设备可以包括键盘、诸如鼠标或轨迹球的定点设备、结合到显示器中的触摸板或触摸屏、滚动轮、点击轮、拨盘、按钮、开关、键盘、具有语音命令识别系统的音频输入设备、麦克风以及其它类型的输入设备。用户接口输入设备可以包括,例如,运动感测和/或手势识别设备,诸如的Microsoft **Kinect®**运动传感器,其使得用户能够使用手势和语音命令通过自然用户接口来控制诸如的Microsoft **Xbox®** 360游戏控制器的输入设备并与之交互。用户接口输入设备也可以包括眼睛姿势识别设备,诸如从用户检测眼睛活动(例如,当拍摄照片和/或做出菜单选择时的“眨眼”)并且将眼睛姿势转换为到输入设备(例如,Google **Glass®**)中的输入的Google **Glass®**眨眼检测器。此外,用户接口输入设备可以包括使用户能够通过语音命令与语音识别系统(例如,**Siri®**导航器)交互的语音识别感测设备。

[0175] 用户接口输入设备也可以包括但不限于三维(3D)鼠标、操纵杆或指向棒、游戏面板和绘图板,以及音频/视频设备,诸如扬声器、数码相机、数码摄像机、便携式媒体播放器、网络摄像头、图像扫描仪、指纹扫描仪、条形码阅读器3D扫描仪、3D打印机、激光测距仪和视

线跟踪设备。此外,用户接口输入设备可以包括,例如,医学成像输入设备,诸如计算机断层扫描、磁共振成像、正电子发射断层摄影术、医疗超声设备。用户接口输入设备也可以包括,例如,诸如MIDI键盘、数字乐器等的音频输入设备。

[0176] 用户接口输出设备可以包括显示子系统、指示灯,或者诸如音频输出设备的非可视显示器,等等。显示子系统可以是阴极射线管(CRT)、诸如使用液晶显示器(LCD)或等离子显示器的平板设备、投影设备、触摸屏,等等。一般而言,术语“输出设备”的使用意在包括用于从计算机系统900向用户或其它计算机输出信息的所有可能类型的设备和机制。例如,用户接口输出设备可以包括,但不限于,可视地传达文本、图形和音频/视频信息的各种显示设备,诸如监视器、打印机、扬声器、耳机、汽车导航系统、绘图仪、语音输出设备,以及调制解调器。

[0177] 计算机系统900可以包括包含软件元件、被示为当前位于系统存储器910中的存储子系统918。系统存储器910可以存储可加载并且可在处理单元904上执行的程序指令,以及在这些程序的执行期间所产生的数据。

[0178] 取决于计算机系统900的配置和类型,系统存储器910可以是易失性的(诸如随机存取存储器(RAM))和/或非易失性的(诸如只读存储器(ROM)、闪存存储器,等等)。RAM通常包含可被处理单元904立即访问和/或目前正被处理单元904操作和执行的数据和/或程序模块。在一些实现中,系统存储器910可以包括多种不同类型的存储器,例如静态随机存取存储器(SRAM)或动态随机存取存储器(DRAM)。在一些实现中,诸如包含有助于在启动期间在计算机系统900的元件之间传送信息的基本例程的基本输入/输出系统(BIOS),通常可以被存储在ROM中。作为示例,但不是限制,系统存储器910也示出了可以包括客户端应用、web浏览器、中间层应用、关系数据库管理系统(RDBMS)等的应用程序912,程序数据914,以及操作系统916。作为示例,操作系统916可以包括各种版本的Microsoft **Windows®**, Apple **Macintosh®** 和/或Linux操作系统、各种可商业获得的 **UNIX®** 或类UNIX操作系统(包括但不限于各种GNU/Linux操作系统、Google **Chrome®**操作系统等)和/或诸如iOS、**Windows®** Phone、**Android®** OS、**BlackBerry®** 100S和 **Palm®** OS操作系统的移动操作系统。

[0179] 存储子系统918也可以提供用于存储提供一些方面的功能的基本编程和数据结构的有形计算机可读存储介质。当被处理器执行时提供上述功能的软件(程序、代码模块、指令)可以被存储在存储子系统918中。这些软件模块或指令可以被处理单元904执行。存储子系统918也可以提供用于存储根据本发明被使用的数据的储存库。

[0180] 存储子系统900也可以包括可被进一步连接到计算机可读存储介质922的计算机可读存储介质读取器920。与系统存储器910一起并且,可选地,与其相结合,计算机可读存储介质922可以全面地表示用于临时和/或更持久地包含、存储、发送和检索计算机可读信息的远程、本地、固定和/或可移除存储设备加存储介质。

[0181] 包含代码或代码的部分的计算机可读存储介质922也可以包括本领域已知或使用的任何适当的介质,包括存储介质和通信介质,诸如但不限于,以用于信息的存储和/或传输的任何方法或技术实现的易失性和非易失性、可移除和不可移除介质。这可以包括有形的非暂时性计算机可读存储介质,诸如RAM、ROM、电可擦除可编程ROM(EEPROM)、闪存存储器

或其它存储器技术、CD-ROM、数字多功能盘 (DVD) 或其它光学存储器、磁带盒、磁带、磁盘存储器或其它磁存储设备, 或者其它有形的计算机可读介质。当被指定时, 这也可以包括非有形的暂时性计算机可读介质, 诸如数据信号、数据传输, 或者可以被用来发送期望信息并且可以被计算机系统900访问的任何其它介质。

[0182] 作为示例, 计算机可读存储介质922可以包括从不可移除的非易失性磁介质读取或写到其的硬盘驱动器、从可移除的非易失性磁盘读取或写到其的磁盘驱动器、以及从可移除的非易失性光盘 (诸如CD ROM、DVD和**Blu-Ray®**盘或其它光学介质) 读取或写到其的光盘驱动器。计算机可读存储介质922可以包括, 但不限于, **Zip®**驱动器、闪存卡、通用串行总线(USB) 闪存驱动器、安全数字(SD) 卡、DVD盘、数字音频带, 等等。计算机可读存储介质922也可以包括基于非易失性存储器的固态驱动器(SSD) (诸如基于闪存存储器的SSD、企业闪存驱动器、固态ROM等)、基于易失性存储器的SSD (诸如固态RAM、动态RAM、静态RAM)、基于DRAM的SSD, 磁阻RAM (MRAM) SSD, 以及使用基于DRAM和闪存存储器的SSD的组的混合SSD。盘驱动器及其关联的计算机可读介质可以为计算机系统900提供计算机可读指令、数据结构、程序模块及其它数据的非易失性存储。

[0183] 通信子系统924提供到其它计算机系统和网络的接口。通信子系统924用作从其它系统接收数据和从计算机系统900向其它系统发送数据的接口。例如, 通信子系统924可以使计算机系统900能够经由互联网连接到一个或多个设备。在一些方面中, 通信子系统924可以包括用于访问无线语音和/或数据网络的射频(RF) 收发器部件 (例如, 使用蜂窝电话技术, 诸如3G、4G或EDGE (用于全球演进的增强型数据速率) 的先进数据网络技术, WiFi (IEEE 802.28系列标准), 或其它移动通信技术, 或其任意组合)、全球定位系统 (GPS) 接收器部件和/或其它部件。在一些方面中, 作为无线接口的附加或者替代, 通信子系统924可以提供有线网络连接 (例如, 以太网)。

[0184] 在一些方面中, 通信子系统924也可以代表可以使用计算机系统900的一个或多个用户接收结构化和/或非结构化数据馈送926、事件流928、事件更新99等形式的输入通信。

[0185] 作为示例, 通信子系统924可被配置为实时地从社交媒体网络和/或其它通信服务的用户接收非结构化的数据馈送926, 诸如**Twitter®**馈送、**Facebook®**更新、诸如丰富站点摘要 (RSS) 馈送的web馈送和/或来自一个或多个第三方信息源的实时更新。

[0186] 此外, 通信子系统924也可被配置为接收连续数据流形式的数据, 这可以包括本质上可以是连续的或无界的没有明确终止的实时事件的事件流928和/或事件更新99。产生连续数据的应用的示例可以包括, 例如, 传感器数据应用、金融报价机、网络性能测量工具 (例如, 网络监视和流量管理应用)、点击流分析工具、汽车流量监视, 等等。

[0187] 通信子系统924也可被配置为向一个或多个数据库输出结构化和/或非结构化数据馈送926、事件流928、事件更新99, 等等, 这一个或多个数据库可以与耦合到计算机系统900的一个或多个流式数据源计算机通信。

[0188] 计算机系统3000可以是各种类型之一, 包括手持便携式设备 (例如, **iPhone®** 蜂窝电话、**iPad®** 计算平板电脑、PDA)、可穿戴设备 (例如, Google **Glass®** 头戴式显示器)、PC、工作站、大型机、信息站、服务器机架、或任何其它数据处理系统。

[0189] 由于计算机和网络的不断变化的本质, 在图中绘出的计算机系统900的描述仅仅

要作为具体的示例。具有比图中绘出的系统更多或更少部件的许多其它配置是可能的。例如,定制的硬件也可以被使用和/或特定的元素可以用硬件、固件、软件(包括applets)或其组合来实现。另外,也可以采用到诸如网络输入/输出设备之类的其它计算设备的连接。基于本文提供的公开内容和示教,本领域普通技术人员将认识到实现各种方面的其它方式和/或方法。

[0190] 在前述的说明书中,本发明的各方面参考其具体方面进行了描述,但本领域技术人员将认识到的是,本发明不限于此。上述发明的各个特征和方面可以被单独或联合使用。此外,在不脱离本说明书的更广泛精神和范围的情况下,方面可以在除本文所述的那些之外的任何数目的环境 and 应用中被使用。相应地,本说明书和附图应当被认为是说明性而不是限制性的。

100

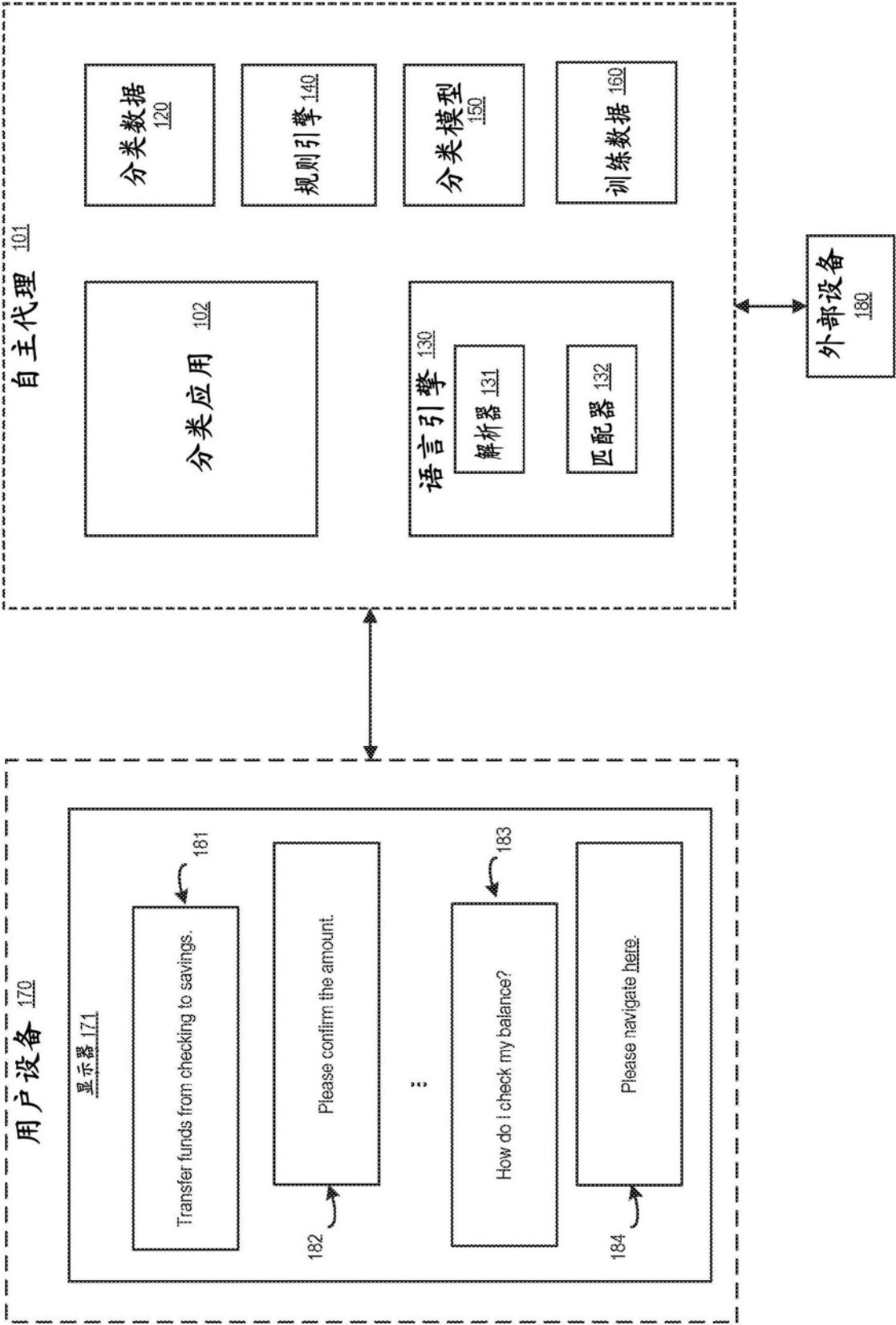


图1

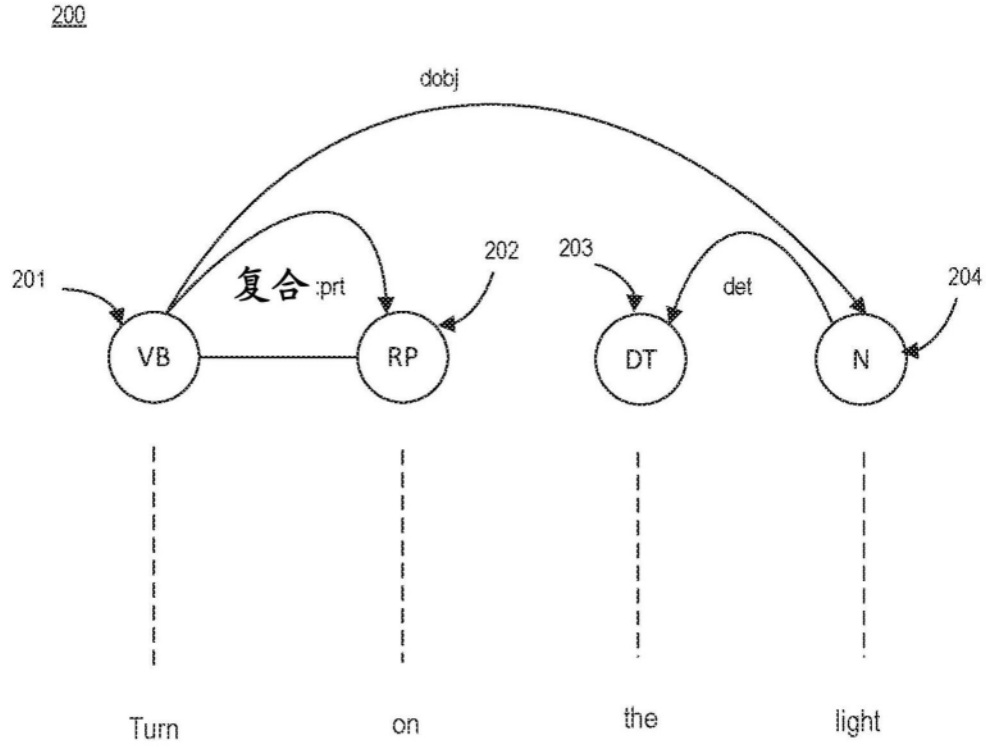


图2

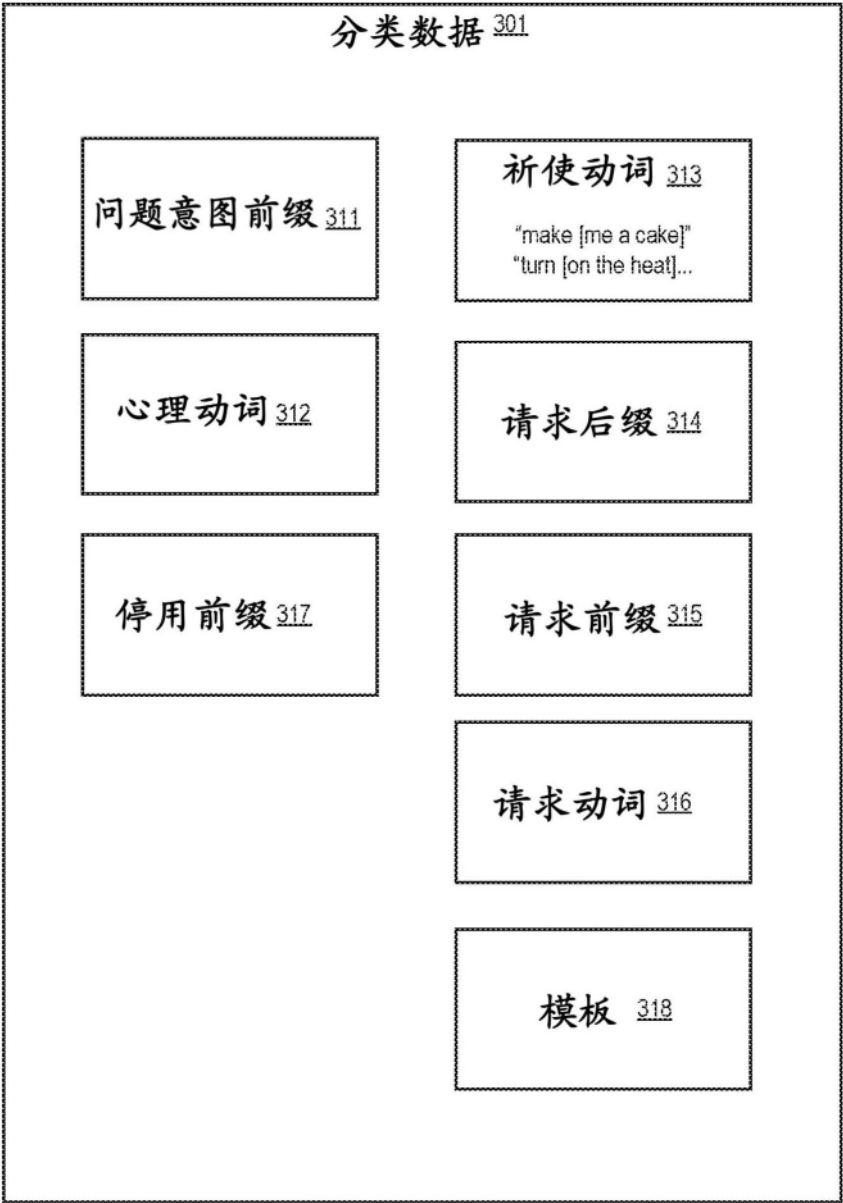


图3

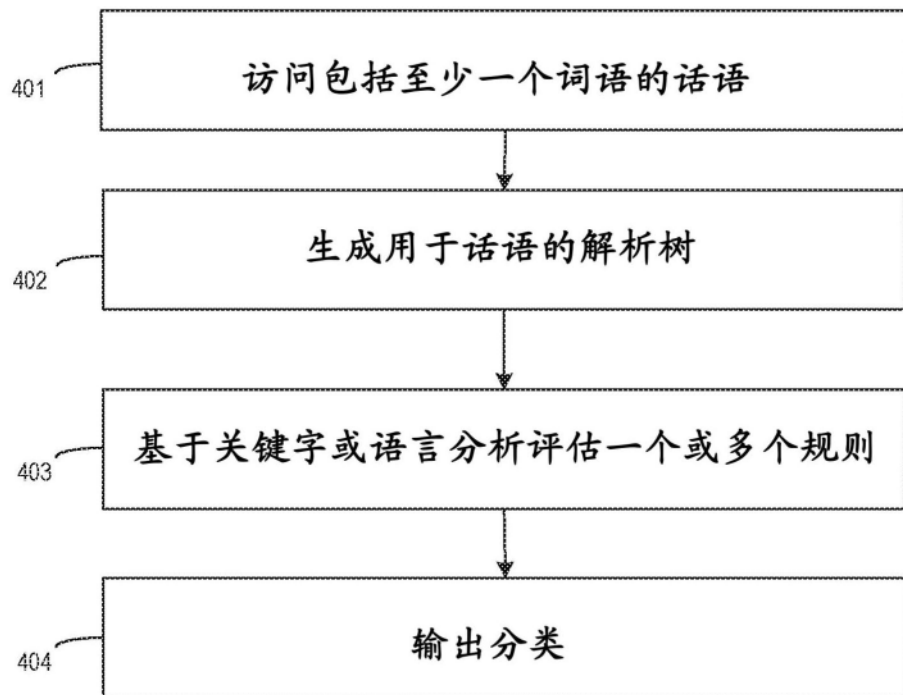
400

图4

500

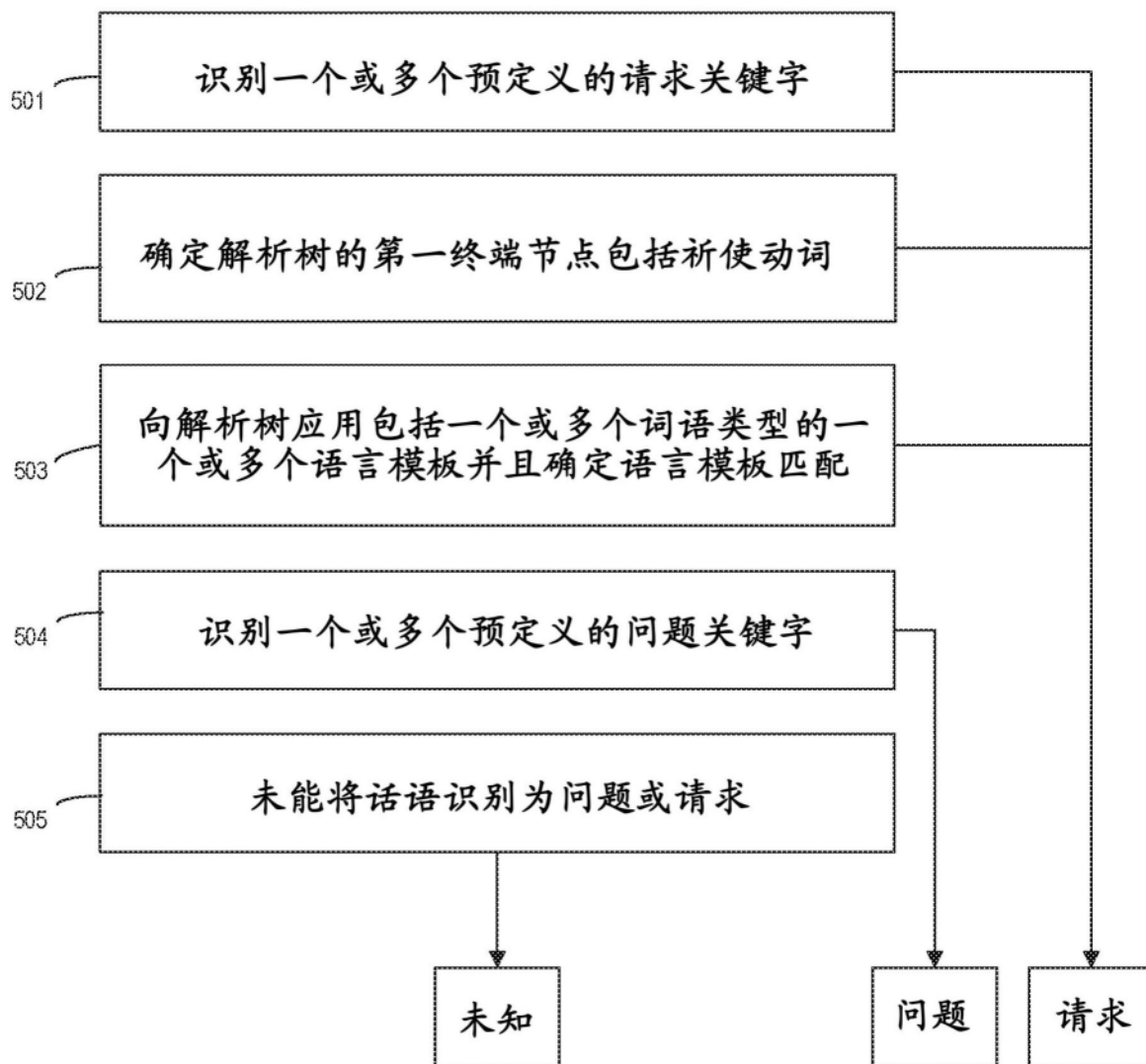


图5

600

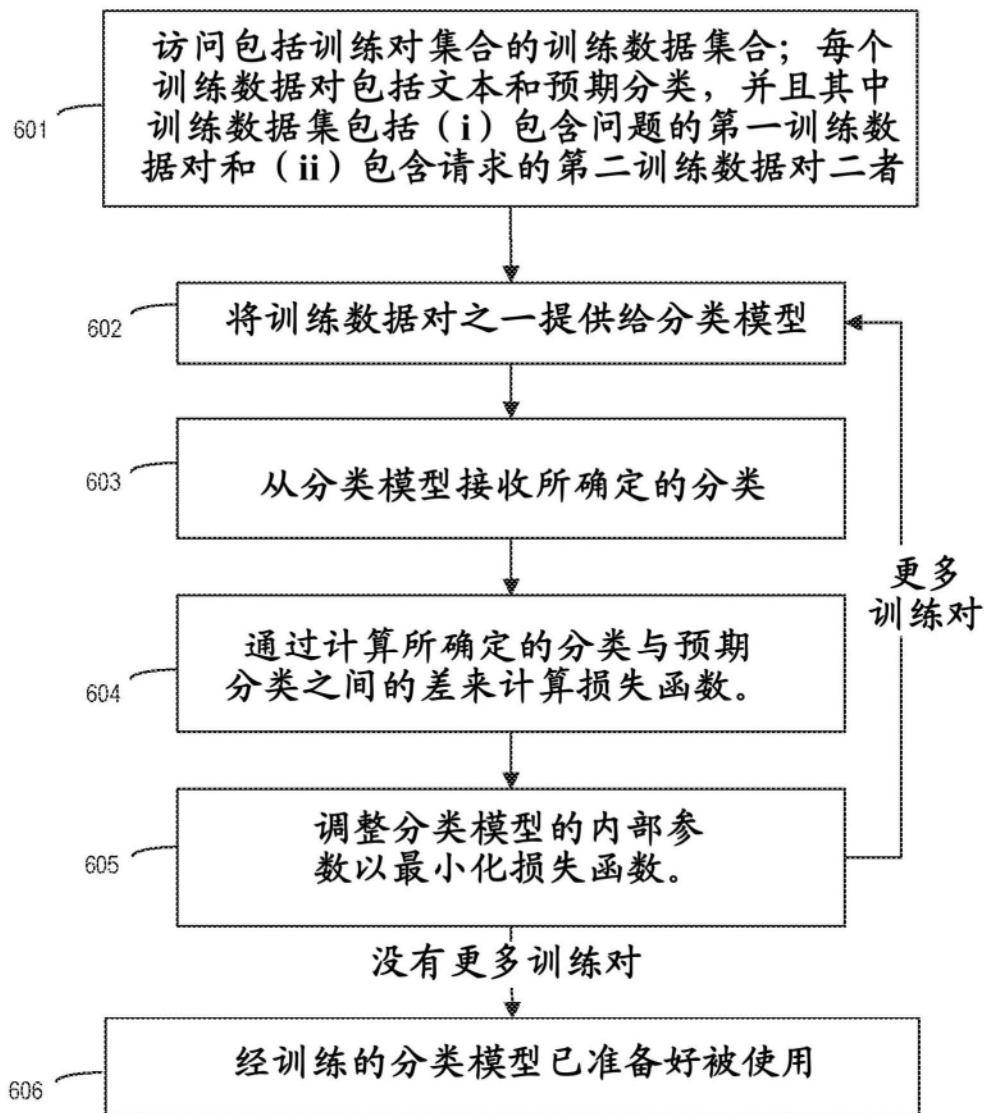


图6

700

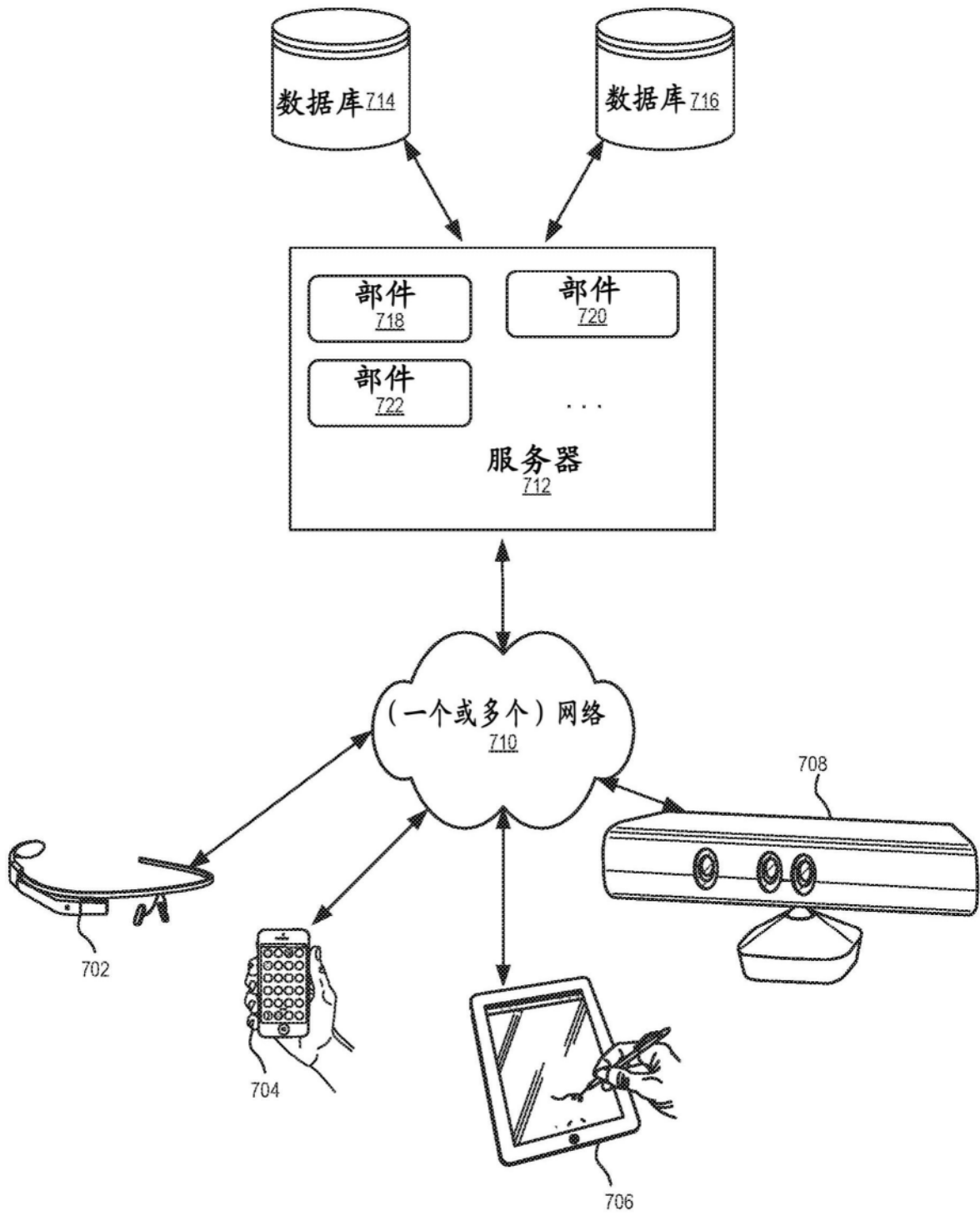


图7

800

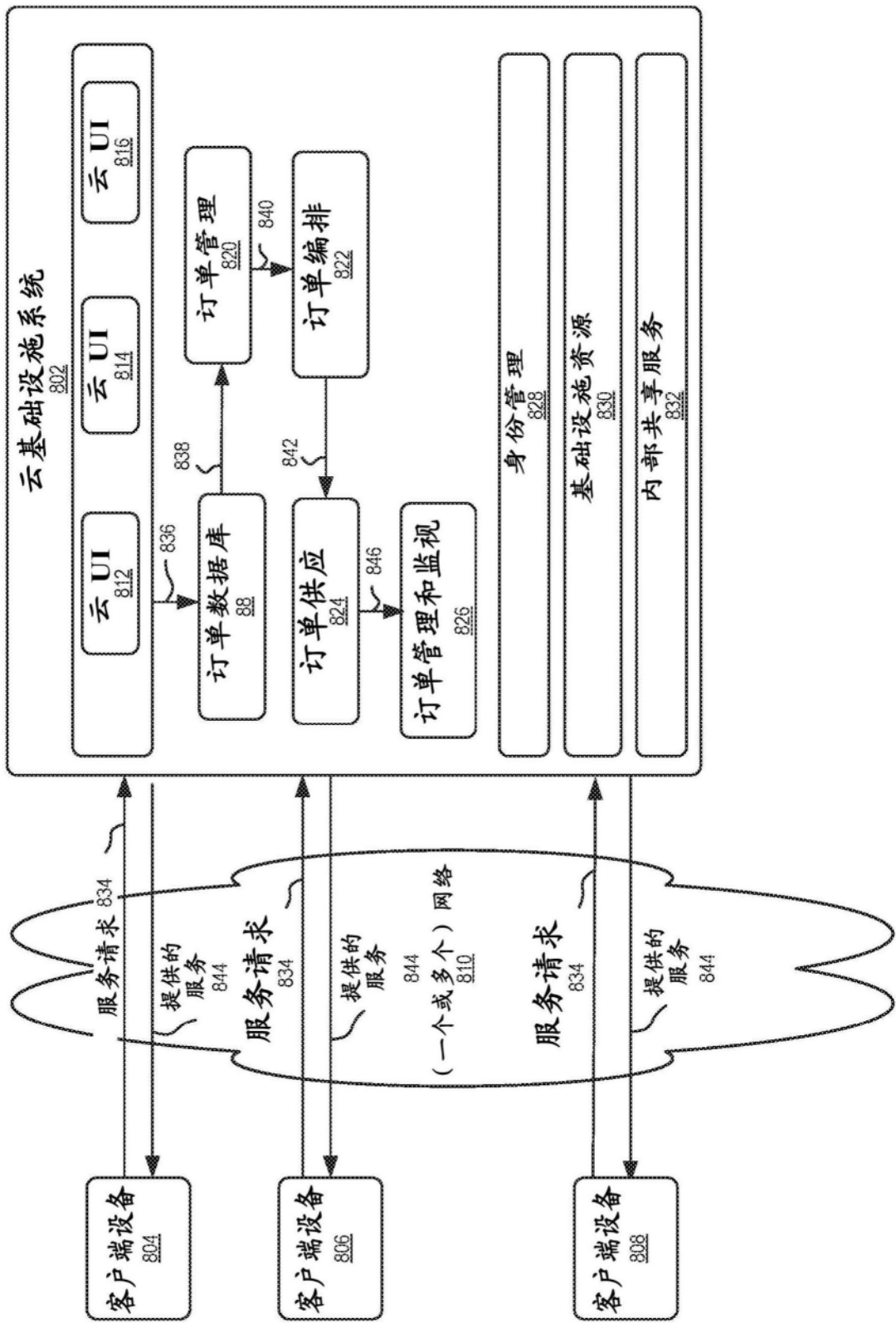


图8

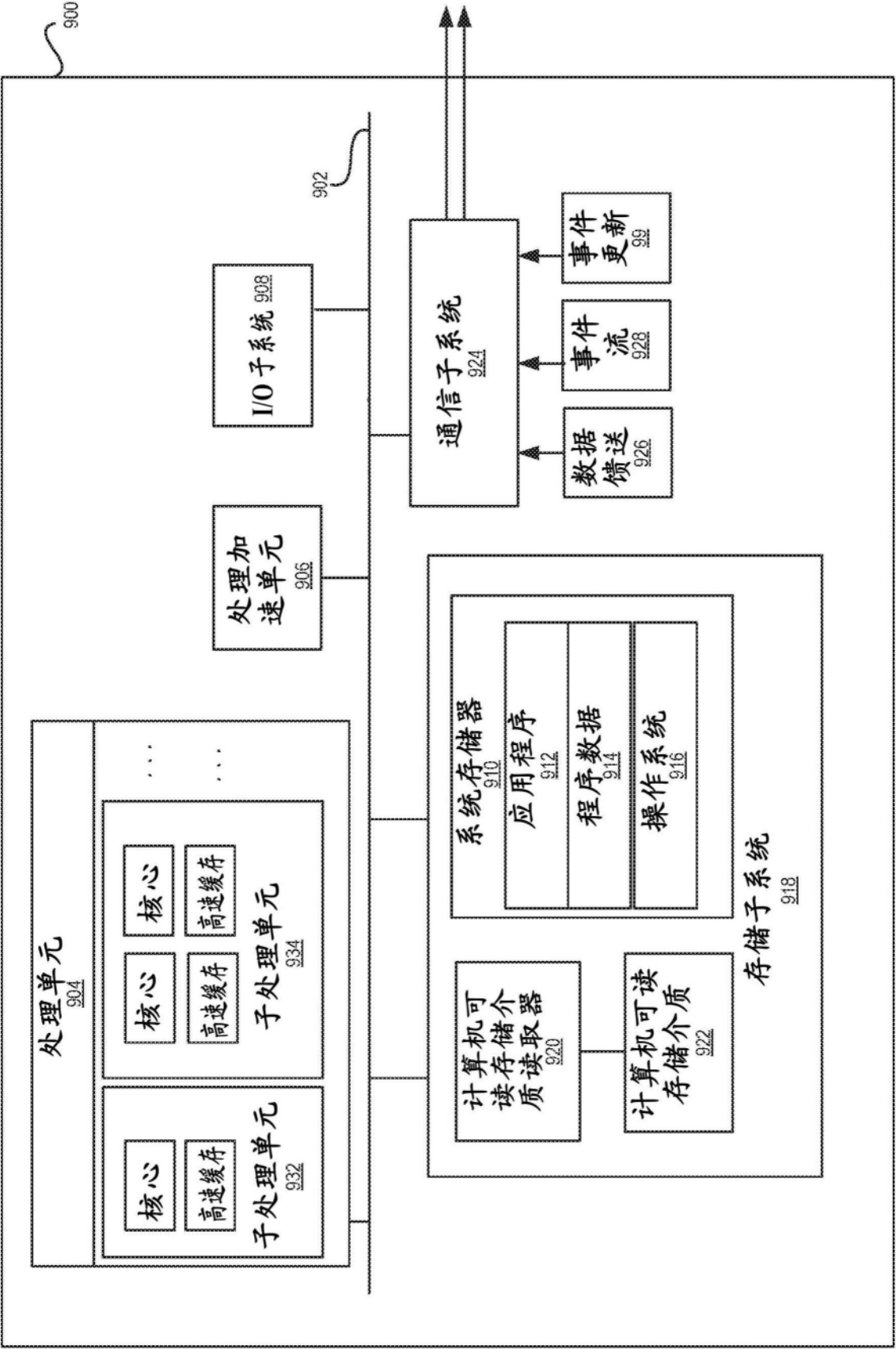


图9