



(12) 发明专利

(10) 授权公告号 CN 111344695 B

(45) 授权公告日 2024.01.23

(21) 申请号 201880072987.4

(74) 专利代理机构 北京市柳沈律师事务所

11105

(22) 申请日 2018.12.12

专利代理人 邝万奎

(65) 同一申请的已公布的文献号

申请公布号 CN 111344695 A

(51) Int.CI.

G06F 16/35 (2019.01)

(43) 申请公布日 2020.06.26

G06F 16/33 (2019.01)

(30) 优先权数据

G06F 16/25 (2019.01)

15/848,563 2017.12.20 US

G06F 16/28 (2019.01)

(85) PCT国际申请进入国家阶段日

G06F 40/30 (2020.01)

2020.05.11

G06F 9/50 (2006.01)

(86) PCT国际申请的申请数据

G06F 18/22 (2023.01)

PCT/IB2018/059902 2018.12.12

G06N 3/0442 (2023.01)

(87) PCT国际申请的公布数据

G06N 3/0455 (2023.01)

W02019/123111 EN 2019.06.27

G06N 3/088 (2023.01)

G06N 3/0895 (2023.01)

(73) 专利权人 国际商业机器公司

(56) 对比文件

地址 美国纽约阿芒克

US 2016267153 A1, 2016.09.15

(72) 发明人 林晨 A.卡利亚 萧晋

US 2015128156 A1, 2015.05.07

M.武科维克 N.阿内罗西斯

CN 102262652 A, 2011.11.30

CN 107479879 A, 2017.12.15

审查员 王东

权利要求书3页 说明书14页 附图10页

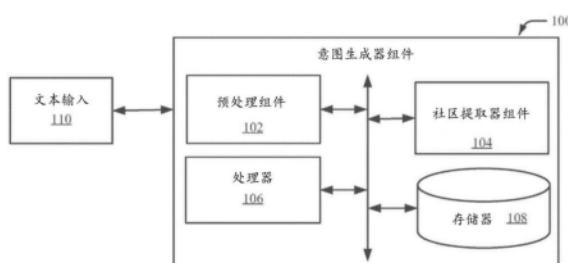
(54) 发明名称

客户端应用程序接口将第一客户端应用程序接口与社区相关联。

促进特定于域和客户端的应用程序接口推荐

(57) 摘要

提供了用于生成特定于域和客户端的应用程序接口推荐的技术。在一个示例中，一种计算机实现的方法包括：第一客户端应用程序接口的第一描述相关联的文本数据而生成的经修改的客户端应用程序接口来确定第一客户端应用程序接口与第二客户端应用程序接口之间的语义相似性。该计算机实现的方法可以进一步包括基于语义相似性将第一描述与第二客户端应用程序接口的第二描述匹配，从而产生一个或多个匹配的客户端应用程序接口。另外，该计算机实现的方法可以进一步包括基于一个或多个匹配的



1. 一种促进应用程序接口推荐的系统,包括:

存储器,存储计算机可执行组件;以及

处理器,执行存储在存储器中的计算机可执行组件,其中计算机可执行组件包括:

预处理器组件,接收与第一客户端应用程序接口的第一描述相关联的第一数据,并通过移除与第一描述相关联的文本数据来预处理第一数据,从而产生经修改的客户端应用程序接口,其中所述文本数据的移除包括移除第一描述中的分别在第一描述中出现至少第一阈值次数的一个或多个术语;以及

意图生成器组件:

确定经修改的客户端应用程序接口与第二客户端应用程序接口之间的语义相似性,其中所述语义相似性基于Louvain社区检测过程;

基于语义相似性将第一描述与第二客户端应用程序接口的第二描述匹配,从而产生一个或多个匹配的客户端应用程序接口;

基于一个或多个匹配的客户端应用程序接口将第一客户端应用程序接口与社区相关联。

2. 如权利要求1所述的系统,其中文本的移除还包括:

移除第一描述中的分别在第一描述中出现少于第二阈值次数的一个或多个附加术语。

3. 如权利要求1所述的系统,其中社区是第一社区,并且意图生成器组件生成包括经修改的客户端应用程序接口的第二社区,其中第二社区用于生成客户端应用程序接口的推荐。

4. 如权利要求1所述的系统,其中计算机可执行组件还包括:

查询匹配器组件,基于语义相似性将第一客户端应用程序接口与一个或多个匹配的客户端应用程序接口进行匹配。

5. 如权利要求1所述的系统,其中,所述社区根据与主题非负矩阵分解相关联的主题建模生成。

6. 如权利要求5所述的系统,其中,所述社区用于生成在提供客户端应用程序接口的推荐时的关键字。

7. 一种促进应用程序接口推荐的计算机可读存储介质,所述计算机可读存储介质具有包含在其中的程序指令,所述程序指令由处理器执行以使处理器:

通过移除与第一描述相关联的文本数据来预处理与第一客户端应用程序接口的第一描述相关联的第一数据,从而产生经修改的客户端应用程序接口,其中所述文本数据的移除包括移除第一描述中的分别在第一描述中出现至少第一阈值次数的一个或多个术语;以及

确定经修改的客户端应用程序接口与第二客户端应用程序接口之间的语义相似性,其中所述语义相似性基于Louvain社区检测过程;

基于语义相似性将第一描述与第二客户端应用程序接口的第二描述匹配,从而产生一个或多个匹配的客户端应用程序接口;和

基于一个或多个匹配的客户端应用程序接口将第一客户端应用程序接口与社区相关联。

8. 如权利要求7所述的计算机可读存储介质,其中所述文本的移除还包括:

移除第一描述中的分别在第一描述中出现少于第二阈值次数的一个或多个附加术语。

9. 如权利要求7所述的计算机可读存储介质,其中社区是第一社区,并且所述程序指令由处理器执行以进一步使处理器生成包括经修改的客户端应用程序接口的第二社区,其中第二社区用于生成客户端应用程序接口的推荐。

10. 如权利要求7所述的计算机可读存储介质,其中程序指令由处理器执行以进一步使处理器:

基于语义相似性将第一客户端应用程序接口与一个或多个匹配的客户端应用程序接口进行匹配。

11. 如权利要求7所述的计算机可读存储介质,其中,所述社区根据与主题非负矩阵分解相关联的主题建模生成。

12. 如权利要求7所述的计算机可读存储介质,其中所述社区用于生成在提供客户端应用程序接口的推荐时的关键字。

13. 一种促进应用程序接口推荐的计算机实现的方法,包括:

由可操作地耦合到处理器的设备通过移除与第一描述相关联的文本数据来预处理与第一客户端应用程序接口的第一描述相关联的第一数据,从而产生经修改的客户端应用程序接口,其中所述文本数据的移除包括移除第一描述中的分别在第一描述中出现至少第一阈值次数的一个或多个术语;以及

由可操作地耦合到处理器的设备确定经修改的客户端应用程序接口与第二客户端应用程序接口之间的语义相似性,其中所述语义相似性基于Louvain社区检测过程;

由所述设备基于语义相似性将第一描述与第二客户端应用程序接口的第二描述匹配,从而产生一个或多个匹配的客户端应用程序接口;和

由所述设备基于一个或多个匹配的客户端应用程序接口将第一客户端应用程序接口与社区相关联。

14. 如权利要求13所述的计算机实现的方法,其中所述文本的移除还包括:

移除第一描述中的分别在第一描述中出现少于第二阈值次数的一个或多个附加术语。

15. 如权利要求13所述的计算机实现的方法,其中社区是第一社区,并且进一步包括由所述设备生成包括经修改的客户端应用程序接口的第二社区,其中第二社区用于生成客户端应用程序接口的推荐。

16. 如权利要求13所述的计算机实现的方法,进一步包括:

由所述设备基于语义相似性将第一客户端应用程序接口与一个或多个匹配的客户端应用程序接口进行匹配。

17. 如权利要求13所述的计算机实现的方法,其中,所述社区根据与主题非负矩阵分解相关联的主题建模生成。

18. 如权利要求13所述的计算机实现的方法,其中,所述社区用于生成在提供客户端应用程序接口的推荐时的关键字。

19. 一种促进应用程序接口推荐的系统,包括:

存储器,存储计算机可执行组件;以及

处理器,执行存储在存储器中的计算机可执行组件,其中可执行组件包括:

查询匹配器组件:

接收包括与应用程序接口描述有关的第一文本数据的查询;和

通过移除第一文本数据中的分别在第一文本数据中出现至少第一阈值次数的一个或多个术语来预处理第一文本数据,从而产生经修改的第一文本数据;和

通过将第一文本数据与第二文本数据进行比较,基于查询的经修改的第一文本数据和与查询相关的主题的第二文本数据之间的相似性来生成分数数据,其中所述相似性基于Louvain社区检测过程。

20. 如权利要求19所述的系统,其中,已确定第二文本数据为指示与主题有关的文本的文本数据。

21. 如权利要求19所述的系统,其中,分数数据包括表示将响应于接收查询而输出的推荐的应用程序接口的推荐的推荐数据。

22. 一种促进应用程序接口推荐的计算机可读存储介质,所述计算机可读存储介质具有包含在其中的程序指令,所述程序指令由处理器执行以使处理器:

接收包括与应用程序接口描述有关的第一文本数据的查询;

通过移除第一文本数据中的分别在第一文本数据中出现至少第一阈值次数的一个或多个术语来预处理第一文本数据,从而产生经修改的第一文本数据;和

通过将第一文本数据与第二文本数据进行比较,基于查询的经修改的第一文本数据和与查询相关的主题的第二文本数据之间的相似性来生成分数数据,其中所述相似性基于Louvain社区检测过程。

23. 如权利要求22所述的计算机可读存储介质,其中已确定第二文本数据为指示与主题有关的文本的文本数据。

促进特定于域和客户端的应用程序接口推荐

背景技术

[0001] 本公开涉及应用程序接口查询，并且更具体地，涉及特定于域和客户端的应用程序接口推荐。

发明内容

[0002] 以下呈现的概述提供对本公开的一个或多个实施例的基本理解。概述并非旨在标识关键或重要单元，也不旨在描绘特定实施例的任何范围或权利要求的任何范围。唯一目的是以促进的形式呈现概念，作为稍后呈现的更详细描述的序言。在本文描述的一个或多个实施例中，描述了促进特定于域和客户端应用程序接口推荐的设备、系统、计算机实现的方法、装置和/或计算机程序产品。

[0003] 根据实施例，系统可以包括：存储器，存储计算机可执行组件；以及处理器，执行存储在存储器中的计算机可执行组件。系统的计算机可执行组件可以包括意图生成器组件，基于通过移除与第一客户端应用程序接口的第一描述相关联的文本数据而生成的经修改的客户端应用程序接口来确定第一客户端应用程序接口与第二客户端应用程序接口之间的语义相似性。意图生成器组件还基于语义相似性将第一描述与第二客户端应用程序接口的第二描述匹配，从而产生一个或多个匹配的客户端应用程序接口。此外，意图生成器组件还基于一个或多个匹配的客户端应用程序接口将第一客户端应用程序接口与社区相关联。

[0004] 根据另一实施例，促进应用程序接口推荐的计算机程序产品可以包括计算机可读存储介质，具有程序指令。程序指令可以由处理器执行，并且处理器可以基于通过移除与第一客户端应用程序接口的第一描述相关联的文本数据而生成的经修改的客户端应用程序接口来确定第一客户端应用程序接口与第二客户端应用程序接口之间的语义相似性。程序指令还可以由处理器执行以基于语义相似性将第一描述与第二客户端应用程序接口的第二描述匹配，从而产生一个或多个匹配的客户端应用程序接口。此外，程序指令还可以由处理器执行以基于一个或多个匹配的客户端应用程序接口将第一客户端应用程序接口与社区相关联。

[0005] 根据又一实施例，提供了一种计算机实现的方法。该计算机实现的方法可以包括：由可操作地耦合到处理器的设备基于通过移除与第一客户端应用程序接口的第一描述相关联的文本数据而生成的经修改的客户端应用程序接口来确定第一客户端应用程序接口与第二客户端应用程序接口之间的语义相似性。该计算机实现的方法还可以包括由设备基于语义相似性将第一描述与第二客户端应用程序接口的第二描述匹配，从而产生一个或多个匹配的客户端应用程序接口。此外，该计算机实现的方法还可以包括由设备基于一个或多个匹配的客户端应用程序接口将第一客户端应用程序接口与社区相关联。

[0006] 根据另一实施例，一种系统可以包括：存储器，存储计算机可执行组件；以及处理器，执行存储在存储器中的计算机可执行组件。该系统的计算机可执行组件可以包括查询匹配器组件，接收包括与应用程序接口描述有关的第一文本数据的查询，并通过将第一文本数据与第二文本数据进行比较，基于查询的第一文本数据和与查询相关的主题的第二文

本数据之间的相似性来生成分数据。

[0007] 根据另一实施例,促进应用程序接口推荐的计算机程序产品可以包括计算机可读存储介质,具有程序指令。程序指令可以由处理器执行,并且处理器可以接收包括与应用程序接口描述有关的第一文本数据的查询,并通过将第一文本数据与第二文本数据进行比较,基于查询的第一文本数据和与查询相关的主题的第二文本数据之间的相似性来生成分数据。

[0008] 在一些实施例中,结合系统、计算机实现的方法和/或计算机程序程序描述的上述元件中的一个或多个可以是实施为诸如计算机实现的方法、计算机程序产品或系统的形式。

附图说明

[0009] 图1示出了根据本文描述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的示例性非限制性系统的框图。

[0010] 图2示出了根据本文描述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的示例性非限制性系统的框图。

[0011] 图3示出了根据本文描述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的示例性非限制性系统的框图。

[0012] 图4示出了根据本文描述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的示例性非限制性系统网络。

[0013] 图5示出了根据本文描述的一个或多个实施例的利用域本体促进查询匹配的示例性非限制性框图。

[0014] 图6示出了根据本文描述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的示例性非限制性流程图。

[0015] 图7示出了根据本文描述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的另一示例性非限制性流程图。

[0016] 图8示出了根据本文描述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的长短期存储器自动编码器的示例性非限制性流程图。

[0017] 图9示出了根据本文描述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的计算机实现的方法的示例性非限制性流程图。

[0018] 图10示出了可以促进本文描述的一个或多个实施例示例非限制性操作环境的框图。

具体实施方式

[0019] 以下详细描述仅是说明性的,并不旨在限制实施例和/或实施例的应用或使用。此外,无意受到前面的背景或摘要部分或详细描述部分中提供的任何明示或暗示信息的约束。

[0020] 现在参考附图描述一个或多个实施例,其中,贯穿全文,相似的附图标记用于指代相似的元件。在以下描述中,出于解释的目的,阐述了许多具体细节,以便提供对一个或多个实施例的更透彻的理解。然而,很明显,在各种情况下,可以在没有这些具体细节的情况下

下实践一个或多个实施例。

[0021] 使用自然语言查询来推荐服务的现有方法是基于有监督和无监督的方法。有监督方法依赖于用特定服务注释的具有自然语言查询的数据集。因为注释过程是手动的，并且通常利用深厚的领域知识，所以这些方法不适用于不同的数据集，尤其是来自新领域的数据集。例如，它们主要基于匹配的关键字、实体关系、主题和聚类。但是，关键字和实体关系忽略了查询和服务之间的语义相似性。主题和聚类捕获语义相似性，但是它们依赖于明确捕获服务之间关系的混搭。但是，对于新领域，此类数据不容易获得。

[0022] 依赖于服务描述来推荐服务的框架可以纠正上述问题。框架的一个或多个实施例的优点是可以立即用作新数据集的引导推荐。为了捕获服务之间的关系，框架的一个或多个实施例可以提供不同的方法来构造社区，其中社区代表对一组服务的抽象。基于社区和用户查询，框架的一个或多个实施例可以应用查询匹配方法来推荐前k个服务。可以针对多个收集的数据集评估框架，以为跨多个收集的数据集的实体提供有意义的推荐。如本文所使用的，术语“实体”可以是或包括硬件、软件、机器、人或其他。

[0023] 本文描述的一个或多个实施例可以生成特定于域和客户端的应用程序接口推荐。另外，本文描述的一个或多个实施例包括促进生成特定于域和客户端的应用程序接口推荐的系统、计算机实施的方法、装置和计算机程序产品。

[0024] 图1示出了根据本文描述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的示例性非限制性系统的框图。

[0025] 如图1所示，意图生成器组件100可以包括几个子组件（例如，预处理器组件102、社区提取器组件104等）、处理器106和存储器108，在各种实施例中，它们可以是电和/或彼此通信地耦合。还应当注意，在一些实施例中，子组件（例如，预处理器组件102、社区提取器组件104等）可以在意图生成器组件100的外部。

[0026] 处理器106的各方面可以构成体现在一个或多个机器中的机器可执行组件，例如，体现在与一个或多个机器关联的一个或多个计算机可读介质（或媒体）中。当由诸如计算机、计算设备、虚拟机等一台或多台机器执行时，可以使机器执行由意图生成器组件100描述的操作。在一方面，意图生成器组件100还可以包括存储计算机可执行组件和指令的存储器108。

[0027] 在一个实施例中，预处理器组件102可以接收自然语言服务描述作为输入，并使用专门的自然语言程序流水线对其进行预处理。预处理可以包括一个或多个功能。例如，可以移除文本输入110的标点符号。另外，可以使用词性（POS）标记器来从文本输入110提取名词短语。在这些步骤之后，可以移除被认为是频繁和/或不频繁的词。因为虽然最频繁的单词可能具有较高的单词计数，但几乎没有语义含义，允许保留频繁的单词常常会降低关联语言模型的有效性。也可以移除最不频繁的单词，因为它们与其他单词同时出现是偶然的。例如，如果服务描述包含对单词“微数据”的单个引用，则“微数据”附近的所有单词最有可能来自一个句子，因此，它们的同时出现不能推广到其他句子。例如，可以移除出现在三个或更多文档中的不频繁的单词和出现在文档总数的百分之十以上的频繁的单词。应当注意，可以使用任何阈值数来移除或提取项，并且上述数仅是示例。

[0028] 主题模型的可解释性可以通过与主题相关的最频繁的术语来判断。基于有效的推理模型，特定于事后语料库的停用词移除可以进一步提高模型的可解释性。特定于语料库

的停用词是出现在许多文档中的术语,但是它们的出现频率不够高,不足以被视为频繁的词。降低频繁词的阈值可能会失去许多其他有用的词。但是,停用词往往会被隔离为一些低质量的主题(例如,不具有连贯和合理意义的词)。因此,可以选择低质量的主题以专门移除主题中最频繁的术语。

[0029] 另外,可以将意图生成器组件100配置为基于应用程序接口(API)的描述确定它们之间的语义相似性。例如,可以经由前面描述的上述预处理器组件102处理来修改与第一客户端应用程序接口相关联的描述,从而产生修改后的API描述。然后可以将修改后的API描述与另一个客户端API描述进行比较,以确定两者之间的相似性。因此,如果满足某个相似性阈值,则意图生成器组件100可以继续进行附加处理。

[0030] 社区提取器组件104可以接收预处理的API描述,并根据它们的相关性形成社区。在每个社区内(或在某些实施例中,在一个或多个社区内),可以构建一个层次结构,其中叶节点是通过主题建模技术推断出的主题,每个主题(叶)(或在一些实施例中,一个或多个主题)可以连接到服务描述列表,其中非叶节点可以表示意图。请注意,结构的深度可以指示不同抽象级别的意图。更接近根节点的意图可以表示抽象意图和一般意图(例如,投资),而更接近主题的意图可以表示详细和特定的意图(例如,证券交易所)。具有共同祖先的主题可以被认为具有较高的相关性。例如,主题“音乐”、“电视”和“视频”之间的联系松散,尽管主题“位置”、“路线”和“地图”可以被认为是紧密相关的。

[0031] 可以通过三种方法来提取社区。第一种是使用主题建模技术的基线方法,其中每个学习的主题(或在某些实施例中,一个或多个学习的主题)形成单个社区。请注意,基线方法不会构造分层意图树。第二种方法可以是一种自下而上的方法,该方法首先推断主题,然后应用社区检测方法来标识那些主题的社区。第三种方法可以是自上而下的方法,该方法学习每个服务(或一个或多个服务)的潜在高级低维矢量表示,然后使用k-均值聚类方法将相似矢量分组为一个社区。此后,可以将主题建模方法应用于每个聚类内(或在某些实施例中,一个或多个聚类内)的服务。

[0032] 主题非负矩阵分解(TNMF)是一种用于从短文本中发现主题的方法。非负矩阵分解(NMF)的传统建模可以分解术语文档矩阵,该矩阵指示术语在给定文档中的出现。但是,对于短文本,术语-文档矩阵可能非常稀疏,这可能会阻止模型学习可靠的主题。TNMF通过假设经常出现的术语最有可能与一个公共主题相关来解决此问题。例如,如果术语“地址”和“邮政编码”同时出现在多个相同的文档中,则他们更有可能谈论一个共同的主题(例如,位置)。因此,TNMF可以通过分解术语相关矩阵来学习可靠的主题。因为此矩阵不会出现稀疏问题,所以即使文档数量增加,词汇量也可以保持稳定。

[0033] 在TNMF中,每个项 t_i 可以表示为术语出现向量 $(w_{i,1}, \dots, w_{i,m})$,其中 $w_{i,m}$ 是根据术语 t_i 和 t_m 的共同出现来确定的。为了评估 $w_{i,m}$,为术语 t_i 和 t_m 计算正点互信息(PPMI)。在计算术语出现向量之后,计算公共向量相似性度量(例如,余弦系数)以产生术语相关矩阵S。一旦学习了术语相关矩阵S,则可以将主题学习问题公式化为查找最小化以下目标函数的术语-主题矩阵U:

[0034] 公式(1): $L(U) = \|S - UU^T\|_F^2$, s. t. $U \geq 0$

[0035] 公式(1)是模型试图最小化的目标函数。这样,公式(1)是非负矩阵分解的公式。F

代表矩阵的Frobenious形式,其中 $\|S - UU^T\|_{ij}^2$ 表示为 $\sum_i (S - UU^T)_{ij}^2$ 。可以将对称非负矩阵分解应用于术语相关矩阵以提取主题并学习术语-主题矩阵U。然后,可以基于原始术语-文档矩阵X和术语-主题矩阵U来求解主题-文档矩阵V。因此,意图生成器组件100可以被配置为接收短文本,通过预处理器组件102运行短文本以提取无关的文本,从而产生精炼的文本数据,然后基于由社区提取器组件104执行TNMF将精炼的文本数据与特定主题相关联。

[0036] 意图生成器组件100可用于使用硬件和/或软件来解决本质上高度技术化的问题(例如,软件文本搜索、语义标记、文本频率的确定、由数百万个术语、句子等组成的语料库中不同术语的匹配),它们不是抽象的,并且由于需要促进用户文本查询中的特定于域和客户端的API推荐所需要的处理能力,因此无法由人类作为一组智力活动来执行。此外,可以由专用计算机执行所执行的一些处理,以执行与存储器操作有关的定义的任务。例如,可以使用专用计算机来基于术语频率等执行与预处理文本输入有关的任务。

[0037] 图2示出了根据本文描述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的示例性非限制性系统的框图。为简洁起见,省略了在此描述的其他实施例中采用的类似元件的重复描述。

[0038] 意图生成器组件200可以包括查询匹配器组件202。查询匹配器组件202可以基于来自实体的查询向实体推荐/输出服务列表。基于查询206,可以对照查询206检查社区和基础主题以确定关系。例如,为了检查相关性,可以在查询206和主题之间生成匹配分数M(查询,主题)(例如,查询中的每个词与给定主题中的每个顶部关键字之间的相似性分数之和)。可以基于Wu-Palmer分数来计算单词相似度,通过考虑与分类法有关的词义深度来计算两个词义的相关性。Wu-Palmer分数还可以计算单词的最不常见的使用者(例如特定祖先节点)的深度。

$$[0039] M(query, topic) = \sum_{u \in query} \sum_{w \in topic} Similarity(u, v)$$

公式(2):

[0040] 公式(3):相似度 $(u, v) = \max(Wu\text{-Palmer}(u_i, v_j))$, (4):其中u是查询206中的单词,ui是主题术语,vj是单词含义。

[0041] 考虑由一组节点和一个根构成的本体。在本体C1和C2中,可以考虑两个本体单元,我们将计算它们的相似度。基于将C1和C2与根节点分开的距离(N1和N2)和将C1和C2的公共最接近祖先与根节点分开的距离(N)计算相似度。相似度定义为Sim_Wu-Palmer,表示为相似度 $(Wu\text{-Palmer}(C1, C2))$ 可以等于 $\frac{2*N}{N1+N2}$ 。就Wordnet而言,一组同义词(同义词集)s1和s2的相似度是通过考虑WordNet分类法中两个同义词集的深度以及最小公有分支(LCS)的深度来计算的。公式为分数 $= \frac{2*depth(LCS)}{depth(s1)+depth(s2)}$ 。M(查询,主题)表示对查询中每个单词u和主题中的单词w的Sim_Wu-Palmer的计算。我们根据查询中的u和主题中的单词w的M(查询,主题)返回的最大值来计算相似度(查询,主题)。

[0042] 因此,如果所考虑的主题属于不同的社区并且不共享同一祖先,则可以标识主题的父节点,并且可以将所标识的意图下的服务作为候选返回。

[0043] 给定一组服务作为候选,可以使用潜在语义索引(LSI)来计算候选服务与查询206

之间的匹配分数。LSI可以通过创建术语频率-逆文档频率(TF-IDF)矩阵并使用奇异值分解(SVD)近似TF-IDF矩阵来创建k维空间。SVD减小了原始矩阵的维数，同时保留了列之间的相似性结构。因此，实体的查询206可以映射到k维LSI空间。可以使用两个向量的余弦相似度来计算查询206和给定文档的相似度。因此，可以将具有最高相似度评分的API服务推荐给实体作为输出208。

[0044] 社区提取器组件104也可以包括神经网络组件204。例如，根据所公开的方面，人工智能(AI)可以促进自动化一个或多个特征。存储器和处理器以及其他组件可以包括关于附图的功能。结合生成API推荐的公开方面可以采用各种基于AI的方案来执行其各个方面。例如，可以通过示例自动分类器系统和过程来促进用于检测一个或多个触发事件，确定作为一个或多个触发事件的结果的活动，以及生成API推荐等的过程。在另一个示例中，可以通过示例自动分类器系统和过程来促进基于频率对某一术语进行惩罚而优选另一个术语的过程。

[0045] 示例性分类器可以是将输入属性向量 $x = (x_1, x_2, x_3, x_4, x_n)$ 映射到输入属于一个类的置信度的函数，即 $f(x) = \text{置信度(类)}$ 。这样的分类可以采用基于概率和/或统计的分析(例如，分解到分析实用程序中)来预测或推断可以自动执行的动作(例如，提取频繁的文本)。支持向量机(SVM)是可以使用的分类器的示例。SVM可以通过在可能的输入空间中找到一个超曲面来进行操作，超曲面会尝试从非触发事件中分离触发条件。凭直觉，这使分类适用于接近但与训练数据不同的测试数据。其他有向和无向的模型分类方法包括，例如，朴素的贝叶斯、贝叶斯网络、递归神经网络、决策树、模糊逻辑模型和提供可采用不同独立性模式的概率分类模型。本文使用的分类还可以包括用于开发优先级模型的统计回归。

[0046] 所公开的方面可以采用被明确训练(例如，经由通用训练数据)以及被隐式训练(例如，经由观察与触发事件有关的术语提取)的分类器。例如，可以在分类器构造函数和特征选择模块内的学习或训练阶段配置SVM。因此，分类器可用于自动学习和执行许多功能，包括但不限于术语提取、术语加权、术语过滤、与主题的术语关联等。该标准可以包括但不限于预定义值、贡献衰减表或其他参数、偏好和/或策略等等。应当注意，神经网络组件204也可以是本公开的各种其他组件的子组件。例如，关于图3讨论的本体组件也可以包括神经网络。

[0047] 图3示出了根据本文所述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的示例性非限制性系统的框图。为简洁起见，省略了在此描述的其他实施例中采用的类似元件的重复描述。

[0048] 在另一个实施例中，意图生成器组件300可以包括本体组件302。本体组件302可以基于特定于客户端和/或特定于领域的查询正式命名和定义主题、属性以及存在的社区和查询的相互关系。本体组件302可以划分意图生成器组件300的计算所需的变量，并建立它们之间的关系。

[0049] 本体组件302可以被配置为基于由查询匹配器组件202生成的匹配、输出208(例如，推荐、置信度分数等)、来自实体的反馈等递归地修改关联的本体。例如，在为输出208生成输出数据之后，可以由本体组件302接收输出数据。本体组件302然后可以从输出数据中生成新的域本体，将新的域本体发送给社区提取器组件104、预处理器组件102和/或查询匹配器组件202。基于本体组件302生成新的域本体，预处理器组件102可以根据新的域本体对

新输入进行预处理以生成其他服务、主题和/或社区。

[0050] 在另一个实施例中，实体可以向意图生成器组件300提供它们自己的特定域本体或偏好，以与意图生成器组件300进行匹配。例如，基于从实体的基于文本的查询生成的输出208（例如，API推荐），实体可以确定实体是否偏好推荐的API或实体是否偏好另一API。因此，与实体的偏好相关联的偏好数据可以被本体组件302接收，以进一步细化匹配过程以用于将来的迭代。因此，在生成其他API推荐时，将来对同一实体的推荐可以考虑实体的偏好。

[0051] 图4示出了根据本文描述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的示例性非限制性系统网络。为简洁起见，省略了在此描述的其他实施例中采用的类似元件的重复描述。

[0052] 在另一实施例中，系统可以采用的一种方法是首先使用TNMF（如上所述）学习主题，然后使用称为Louvain社区检测（LCD）的贪婪优化算法，使用对主题和服务之间成对关联建模的构建网络400提取社区。更具体地，基于学习的主题-文档矩阵V来构建网络400。在网络400中，节点可以表示服务（或文档）和主题。如果V中的相应条目不为零，则可以在主题和服务之间形成加权边。因此，权重就是条目中的值。

[0053] 如图所示，在完成网络400之后，可以应用LCD来从中提取社区。LCD是一种贪婪的优化算法，其中LCD要优化的值是一个称为模块化的度量值（如图所示介于-1到1之间）。模块化度量社区内部边缘到社区外部边缘的密度。因此，优化模块化可以导致社区的分裂。遍历所有可能的分区是非常不切实际的。因此，可以采用启发式方法，其中以下两个阶段迭代地重复进行，直到不发生任何变化为止。

[0054] 在第一阶段，将所有节点分配给它们自己的社区。为将节点i从其社区402移动到相邻社区404j（在开始时仅包含节点j）计算模块化的变化。可以使用以下启发式函数来计算模块化的变化：

$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right]$$

公式(4):

[0056] 其中 \sum_{in} 是i要进入的社区(C)（例如，社区404）内边缘的所有权重的总和（在某些实施例中，可以将 \sum_{in} 视为代表C内链接的总和）； \sum_{tot} 表示入射到C中的节点的链接的总和。 \sum_{tot} 是社区（例如，社区402）中边缘的所有权重的总和。变量 k_i 是i的加权度（在某些实施例中， k_i 可被认为是入射到节点i的链接的权重之和）， k_i, in 是i与社区（例如，社区402）中其他节点之间的链接的权重之和， m 是网络中所有链接的权重之和。

[0057] 在一些实施例中， \sum_{in} 表示C内的链接的总和； \sum_{tot} 表示入射到C中节点的链接的总和； k_i 是入射到节点i的链接权重的总和； k_i, in 是从i到C中节点的链接权重之和； m 是网络中所有链接的权重之和。

[0058] 一旦为节点i连接的所有社区计算了值，就可以将节点i组合到社区404中，从而得到最大的 ΔQ 。此过程可以顺序应用于网络中的所有社区。在第二阶段中，可以将同一社区中的节点分组在一起，并且可以将整个社区视为一个节点，从而可以构建新的网络。然后，可以重新应用第一阶段，并且可以重复此过程，直到模块化的变化不大于定义的阈值为止。

[0059] 图5示出了根据本文所述的一个或多个实施例的促进利用域本体进行查询匹配的示例性非限制性框图。为简洁起见，省略了在此描述的其他实施例中采用的类似元件的重复描述。

[0060] 基于包含文本502(例如“地图,位置,地址”)的查询206,可以针对查询206检查社区和基础主题以确定关系。例如,为了检查单词“位置”的查询206的相关性,可以使用所提供的相似术语(例如,“地图”和“地址”)来生成匹配分数M(查询,主题)。另外,可以在查询206和主题之间生成匹配分数M(例如,查询中的每个单词与给定主题中的每个顶部关键字之间的相似性分数之和)。因此,可以将单词“地图”,“位置”和“地址”确定为与主题1相关,其中在社区发现过程504中确定主题1与社区1相关。此后,潜在语义索引(可以基于与社区1相关的特定文档执行LSI)以识别候选API 506。

[0061] 因此,给定一组服务作为候选,LSI可用于计算候选服务和查询之间的匹配分数。LSI通过创建术语频率-逆文档频率(TF-IDF)矩阵并使用奇异值分解(SVD)近似(TF-IDF)矩阵来创建k维空间。SVD减小了原始矩阵的维数,同时保留了列之间的相似性结构。因此,实体的查询206被映射到k维LSI空间。因此,可以使用两个向量的余弦相似度来计算查询206和给定文档的相似度。

[0062] 图6示出了根据本文描述的一个或多个实施例的示例性的促进特定于域和客户端的应用程序接口推荐的非限制性流程图。为简洁起见,省略了在此描述的其他实施例中采用的类似元件的重复描述。

[0063] 在另一实施例中,为了学习分层意图,API推荐过程600可以包括在框602接收与API相关联的术语。例如,可以将API的数据存储、API描述和/或与API相关联的术语上传到意图生成器组件100、200、300。然后,意图生成器组件100、200、300可以开始通过预处理器组件102对上传的API术语进行预处理。可以通过几种不同的方法对API术语进行预处理,包括但不限于:1) 移除标点并提取词性标签,2) 过滤频繁和/或不频繁的术语,3) 移除规范停用词,和/或4) 提取和移除重复但无用的频繁词。在一实施例中,TNMF可用于提取和移除重复但无信息的频繁词。

[0064] 在执行预处理之后,意图生成器组件100、200、300可以在框606决定采用两种方法中的一种来基于术语相关性形成社区。第一种方法包括在框608将TNMF和社区检测(例如,关于图4讨论的Louvain的社区检测(LCD))应用于预处理后的术语。第二种方法包括在框610应用长短期记忆(LSTM)、k-均值聚类和对预处理术语的主题建模(如关于图8所讨论的)。来自框608、610的输出都可以用于生成关键字并在框612为API分配社区。还应注意,在其他实施例中,TNMF可用于形成社区而无需使用LCD方法。

[0065] 图7示出了根据本文所述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的另一示例性非限制性流程图。为简洁起见,省略了在此描述的其他实施例中采用的类似元件的重复描述。

[0066] 在框702,可以从实体接收针对API推荐的基于文本的查询。基于文本的查询可以包括与感兴趣的API相关联的描述。例如,实体可能正在寻找与位置相关联的API。在框704,可以从描述中提取目标词和动词,以基于文本的查询来确定实体的预测意图。基于在框612每个聚类或社区(通过社区提取器组件104)生成的关键字,在框708,实体的预测意图可以与参考先前讨论的分层意图学习过程生成的分层意图匹配。实体意图与分层意图的匹

配可以基于匹配的意图产生用于推荐与框710的位置相关联的API的数据。

[0067] 意图生成器组件100、200、300可被配置为基于推荐的API提示实体并从实体接收反馈，并且在框712收集实体反馈。例如，在框710推荐了API之后意图生成器组件100、200、300可以提示实体以获取有关推荐API的反馈。反馈可以代表根据实体发起的查询，实体是否发现有用的API。反馈可以包括以下指示：推荐的API不适用，推荐的API是实体所偏爱的，或者推荐的API仅包括实体所偏爱的一些属性。然后，收集的实体反馈可以被发送到框714，可以被用于确定特定于客户端或域的本体。例如，如果实体反馈指示所推荐的API与实体所查询的API相距甚远，则意图生成器组件300可以确定实体可能具有自己的本体，将来可以应用于实体中以使意图生成器组件100、200、300可以提供更准确的推荐结果。因此，将来，特定于客户端或域的本体相关联的数据也可以用于将实体的意图与分层意图进行匹配。

[0068] 图8示出了根据本文描述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的长短期存储器自动编码器的示例性非限制性流程图。为简洁起见，省略了在此描述的其他实施例中采用的类似元件的重复描述。

[0069] 在另一个实施例中，可以训练基于长短期存储器(LSTM)的自动编码器以建立用于服务描述的嵌入，然后解码该嵌入以重建原始句子。嵌入可视为原始服务描述的高级低维度表示。此外，k-均值聚类方法可用于将嵌入服务的分区划分为不同的聚类或社区。

[0070] LSTM包括捕获单词之间的局部依赖性的能力，其中相邻单词被组合以表达特定含义。LSTM是一种递归神经网络(RNN)，可以避免消失(和爆炸)的梯度问题。LSTM单元可以包含三个组件：遗忘、输入和输出。这些组件彼此交互以控制信息的流动方式。遗忘组件确定来自先前存储单元的哪些信息(由于过期)不再有效而应丢弃。输入组件确定哪些信息是新信息并需要更新，输出组件是激活功能，可过滤存储单元中的值。

[0071] 例如，如图8所示，单元802可以包括输入可以在单元804编码的文本。自动编码器可以利用无监督(或自监督)技术来学习服务描述的紧凑表示而不需要其他标签。基于LSTM的自动编码器可以是包含两个LSTM的神经模型，其中一个LSTM可以将单词序列编码为固定长度的矢量表示，从而在单元806生成压缩表示。最后一个LSTM单元的输出可以生成服务描述的嵌入。在单元808，第二LSTM可以将表示解码为另一符号序列。可以共同训练所提出模型的编码器和解码器，以在给定源序列的情况下最大化目标序列的条件概率，从而在单元810生成目标。在获得低维表示之后，可以采用k-均值聚类方法。在单元812应用“服务”将服务划分为k个指定的聚类或社区。最后，可以在单元814将TNMF应用于每个社区内的服务以提取每个社区的主题。

[0072] 图9示出了根据本文所述的一个或多个实施例的促进特定于域和客户端的应用程序接口推荐的计算机实现的方法900的示例性非限制性流程图。为简洁起见，省略了在此描述的其他实施例中采用的类似元件的重复描述。

[0073] 在另一个实施例中，计算机实现的方法900包括：通过可操作地耦合到处理器的设备，通过移除与客户端应用程序接口的描述相关联的文本数据(例如，经由预处理器组件102)修改客户端应用程序接口的描述，得到修改的客户端应用程序接口的描述。计算机实现的方法可以进一步包括基于修改的客户端应用程序接口的描述和一个或多个先前客户端应用程序接口的一个或多个先前描述之间的语义相似性(例如，经由本体组件302)分析与客户端应用程序接口相关联的性能以(例如，经由本体组件302)生成本体。

[0074] 为了提供所公开的主题的各个方面的上下文,图10以及以下讨论旨在提供可以在其中实施所公开的主题的各个方面的合适环境的一般描述。图10示出了可以促进本文所述的一个或多个实施例的示例非限制性操作环境的框图。参考图10,用于实现本公开的各个方面的合适的操作环境1000也可以包括计算机1012。计算机1012还可以包括处理单元1014、系统存储器1016和系统总线1018。系统总线1018将包括但不限于系统存储器1016的系统组件耦合到处理单元1014。处理单元1014可以是各种可用处理器中的任何一个。双微处理器和其他多处理器体系结构也可以用作处理单元1014。系统总线1018可以是几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、外围总线或外部总线和/或使用任何可用总线架构的本地总线,包括但不限于工业标准架构 (ISA)、微通道架构 (MSA)、扩展ISA (EISA)、智能驱动电子 (IDE)、VESA本地总线 (VLB)、外围组件互连 (PCI)、卡总线、通用串行总线 (USB)、高级图形端口 (AGP)、火线 (IEEE 1394) 和小型计算机系统接口 (SCSI)。

[0075] 系统存储器1016还可以包括易失性存储器1020和非易失性存储器1022。包含诸如在启动期间在计算机1012内的元件之间传递信息的基本例程的基本输入/输出系统 (BIOS) 存储在非易失性存储器1022中。作为示例而非限制,非易失性存储器1022可以包括只读存储器 (ROM)、可编程ROM (PROM)、电可编程ROM (EPROM)、电可擦除可编程ROM (EEPROM)、闪存或非易失性随机存取存储器 (RAM) (例如铁电RAM (FeRAM))。易失性存储器1020还可以包括用作外部高速缓冲存储器的随机存取存储器 (RAM)。作为说明而非限制, RAM有多种形式,例如静态RAM (SRAM)、动态RAM (DRAM)、同步DRAM (SDRAM)、双倍数据速率SDRAM (DDR SDRAM)、增强型SDRAM (ESDRAM)、同步链接DRAM (SLDRAM)、直接Rambus RAM (DRRAM)、直接Rambus动态RAM [0076] (DRDRAM) 和Rambus动态RAM。

[0077] 计算机1012还可以包括可移动/不可移动、易失性/非易失性计算机存储介质。图10举例说明了磁盘存储设备1024。磁盘存储设备1024还可以包括但不限于磁盘驱动器、软盘驱动器、磁带驱动器、Jaz驱动器、Zip驱动器、LS-100驱动器、闪存卡或记忆棒。磁盘存储设备1024还可以单独包含存储介质,也可以与其他存储介质结合使用,包括但不限于光盘驱动器,例如光盘ROM设备 (CD-ROM)、可记录CD的驱动器 (CD-R驱动器)、CD可重写驱动器 (CD-RW驱动器) 或数字多功能磁盘ROM驱动器 (DVD-ROM)。为了促进磁盘存储1024与系统总线1018的连接,通常使用可移动或不可移动的接口,例如接口1026。图10还描绘了充当用户与适当的操作环境1000中描述的基本计算机资源之间的中介的软件。这样的软件还可以包括例如操作系统1028。可以存储在磁盘存储器1024上的操作系统1028用于控制和分配计算机1012的资源。

[0078] 系统应用1030利用操作系统1028通过程序模块1032和程序数据1034(例如,存储在系统存储器1016或磁盘存储1024中的程序数据1034)对资源的管理。应当理解,本公开可以是用各种操作系统或操作系统组合实现。用户通过输入设备1036向计算机1012输入命令或信息。输入设备1036包括但不限于指示设备,例如鼠标、轨迹球、触控笔、触摸板、键盘、麦克风、操纵杆、游戏垫、卫星天线、扫描仪、电视调谐卡、数码相机、数码摄像机、网络摄像机等。这些输入设备和其他输入设备通过系统总线1018经由接口端口1038连接到处理单元1014。接口端口1038例如包括串行端口、并行端口、游戏端口和通用串行总线 (USB)。输出设备1040使用与输入设备1036相同类型的端口中的一些。因此,例如,USB端口可以用于向计算机1012提供输入,以及将信息从计算机1012输出到输出。提供输出适配器1042以说明存

在一些输出设备1040,例如监视器、扬声器和打印机,以及需要特殊适配器的其他输出设备1040。作为说明而非限制,输出适配器1042包括视频和声卡,提供输出设备1040和系统总线1018之间的连接方式。应当注意,其他设备和/或设备系统同时提供输入和输出功能,例如远程计算机1044。

[0079] 计算机1012可以使用到诸如远程计算机1044的一个或多个远程计算机的逻辑连接来在联网环境中操作。远程计算机1044可以是计算机、服务器、路由器、网络PC、工作站、基于微处理器的设备、对等设备或其他公共网络节点等,并且通常还可以包括相对于计算机1012的许多或所有单元。为简洁起见,仅示出了存储器存储设备1046具有远程计算机1044。远程计算机1044通过网络接口1048逻辑地连接到计算机1012,然后经由通信连接1050物理地连接。网络接口1048包括诸如本地的有线和/或无线通信网络,例如区域网络(LAN)、广域网(WAN)、蜂窝网络等。LAN技术包括光纤分布式数据接口(FDDI)、铜缆分布式数据接口(CDDI)、以太网、令牌环网等。WAN技术包括但不限于点对点链接、电路交换网络(例如集成服务数字网络(ISDN))及其变体、分组交换网络和数字用户线(DSL)。一个或多个通信连接1050是指用于将网络接口1048连接到系统总线1018的硬件/软件。尽管为了说明清楚起见,在计算机1012内部示出了通信连接1050,但是它也可以在计算机1012的外部。仅用于示例性目的,用于连接到网络接口1048的软件还可以包括内部和外部技术,例如包括常规电话级调制解调器、电缆调制解调器和DSL调制解调器的调制解调器、ISDN适配器和以太网卡。

[0080] 在任何可能的技术细节结合层面,本公开可以是系统、方法和/或计算机程序产品。计算机程序产品可以包括计算机可读存储介质,其上载有用于使处理器实现本公开的各个方面的计算机可读程序指令。计算机可读存储介质可以是可以保持和存储由指令执行设备使用的指令的有形设备。计算机可读存储介质例如可以是一一但不限于一一电存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或者上述的任意合适的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括:便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、静态随机存取存储器(SRAM)、便携式压缩盘只读存储器(CD-ROM)、数字多功能盘(DVD)、记忆棒、软盘、机械编码设备、例如其上存储有指令的打孔卡或凹槽内凸起结构、以及上述的任意合适的组合。这里所使用的计算机可读存储介质不被解释为瞬时信号本身,诸如无线电波或者其他自由传播的电磁波、通过波导或其他传输媒介传播的电磁波(例如,通过光纤电缆的光脉冲)、或者通过电线传输的电信号。

[0081] 这里所描述的计算机可读程序指令可以从计算机可读存储介质下载到各个计算/处理设备,或者通过网络、例如因特网、局域网、广域网和/或无线网下载到外部计算机或外部存储设备。网络可以包括铜传输电缆、光纤传输、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配卡或者网络接口从网络接收计算机可读程序指令,并转发该计算机可读程序指令,以供存储在各个计算/处理设备中的计算机可读存储介质中。用于执行本发明操作的计算机程序指令可以是汇编指令、指令集架构(ISA)指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、集成电路配置数据或者以一种或多种编程语言的任意组合编写的源代码或目标代码,所述编程语言包括面向对象的编程语言—诸如Smalltalk、C++等,以及过程式编程语言—诸如“C”语言或类似的

编程语言。计算机可读程序指令可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络—包括局域网（LAN）或广域网（WAN）—连接到用户计算机，或者，可以连接到外部计算机（例如利用因特网服务提供商来通过因特网连接）。在一些实施例中，通过利用计算机可读程序指令的状态信息来个性化定制电子电路，例如可编程逻辑电路、现场可编程门阵列（FPGA）或可编程逻辑阵列（PLA），该电子电路可以执行计算机可读程序指令，从而实现本发明的各个方面。

[0082] 这里参照根据本公开实施例的方法、装置（系统）和计算机程序产品的流程图和/或框图描述了本公开的各个方面。应当理解，流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合，都可以由计算机可读程序指令实现。这些计算机可读程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器，从而生产出一种机器，使得这些指令在通过计算机或其它可编程数据处理装置的处理器执行时，产生了实现流程图和/或框图中的一个或多个方框中规定的功能/动作的装置。也可以把这些计算机可读程序指令存储在计算机可读存储介质中，这些指令使得计算机、可编程数据处理装置和/或其他设备以特定方式工作，从而，存储有指令的计算机可读介质则包括一个制造品，其包括实现流程图和/或框图中的一个或多个方框中规定的功能/动作的各个方面的指令。也可以把计算机可读程序指令加载到计算机、其它可编程数据处理装置、或其它设备上，使得在计算机、其它可编程数据处理装置或其它设备上执行一系列操作步骤，以产生计算机实现的过程，从而使得在计算机、其它可编程数据处理装置、或其它设备上执行的指令实现流程图和/或框图中的一个或多个方框中规定的功能/动作。

[0083] 附图中的流程图和框图显示了根据本公开的多个实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段或指令的一部分，所述模块、程序段或指令的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。在有些作为替换的实现中，方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个连续的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意的是，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行规定的功能或动作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

[0084] 尽管以上已经在运行于一个或多个计算机上的计算机程序产品的计算机可执行指令的一般上下文中描述了主题，但是本领域技术人员将认识到，本公开内容也能够以与其他程序模块组合的方式实现。通常，程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、组件、数据结构等。此外，本领域技术人员将理解，可以利用其他计算机系统配置来实践本发明的计算机实现的方法，包括单处理器或多处理器计算机系统、小型计算设备、大型计算机以及手持式计算机、计算设备（例如PDA，电话）、基于微处理器或可编程的消费类或工业电子产品等。所说明的方面也可以在分布式计算环境中实践。在分布式计算环境中，任务由通过通信网络链接的远程处理设备执行。然而，本公开的一些（如果不是全部）方面可以在独立计算机上实践。在分布式计算环境中，程序模块可以位于本地和远程存储设备中。

[0085] 如在本申请中使用的,术语“组件”、“系统”、“平台”、“接口”等可以指代或可以包括具有一个或多个特定功能的计算机相关实体或与操作机器相关的实体。本文公开的实体可以是硬件、硬件和软件的组合、软件或执行中的软件。例如,组件可以是但不限于在处理器上运行的进程、处理器、对象、可执行文件、执行线程、程序或计算机。作为说明,在服务器上运行的应用程序和服务器都可以是组件。一个或多个组件可以驻留在执行的进程或线程中,并且一个组件可以位于一台计算机上或分布在两台或多台计算机之间。在另一个示例中,各个组件可以从其上存储有各种数据结构的各种计算机可读介质执行。这些组件可以例如根据具有一个或多个数据包的信号(例如,来自与本地系统、分布式系统或跨网络(例如Internet)的一个组件通过信号交互的另一组件的数据)经由本地或远程进程通信。作为另一示例,组件可以是具有由电气或电子电路操作的机械部件提供的特定功能的设备,由处理器执行的软件或固件应用程序操作。在这种情况下,处理器可以在设备内部或外部,并且可以执行软件或固件应用程序的至少一部分。作为又一个示例,组件可以是通过不具有机械部件的电子组件提供特定功能的装置,其中电子组件可以包括处理器或其他装置,以执行至少部分地赋予电子组件功能的软件或固件。在一方面,组件可以经由例如云计算系统内的虚拟机来仿真电子组件。

[0086] 另外,术语“或”旨在表示包含性“或”而不是排他性“或”。也就是说,除非另有说明或从上下文可以清楚地看出,否则“X使用A或B”旨在表示任何自然的包含性排列。也就是说,如果X使用A,X使用B;或X使用A和B两者,则在任何上述情况下均满足“X使用A或B”。此外,在主题说明书和附图中使用的冠词“一”和“一个”通常应被解释为意指“一个或多个”,除非另有说明或从上下文清楚地理解为是单数形式。如本文所使用的,术语“示例”或“示例性”用于表示用作示例、实例或说明。为了避免疑问,本文所公开的主题不受这些示例的限制。另外,本文描述为“示例”或“示例性”的任何方面或设计不必被解释为比其他方面或设计更优选或有利,也不意味着排除本领域普通技术人员已知的等同示例性结构和技术。

[0087] 如在本主题说明书中采用的,术语“处理器”可以基本上指包括但不限于单核处理器的任何计算处理单元或设备、具有软件多线程执行能力的单处理器、多核处理器、具有软件多线程执行能力的多核处理器、具有硬件多线程技术的多核处理器、并行平台和具有分布式共享内存的并行平台。另外,处理器可以指集成电路、专用集成电路(ASIC)、数字信号处理器(DSP)、现场可编程门阵列(FPGA)、可编程逻辑控制器(PLC)、复杂的可编程逻辑设备(CPLD)、分立的栅极或晶体管逻辑、分立的硬件组件或其任意组合设计成执行本文所述的功能。此外,处理器可以利用纳米级架构,例如但不限于基于分子和量子点的晶体管、开关和门,以优化空间使用或增强用户设备的性能。处理器也可以被实现为计算处理单元的组合。在本公开中,诸如“存储”、“储存”、“数据存储”、“数据储存”、“数据库”以及与组件的操作和功能有关的任何其他信息存储组件的术语被用来指代“存储器组件”,体现在“内存”中的实体或包含存储器的组件。应当理解,本文所述的存储器或存储器组件可以是易失性存储器或非易失性存储器,或者可以包括易失性和非易失性存储器两者。作为说明而非限制,非易失性存储器可以包括只读存储器(ROM)、可编程ROM(PROM)、电可编程ROM(EPROM)、电可擦除ROM(EEPROM)、闪存或非易失性随机存取存储器(RAM)(例如铁电RAM(FeRAM))。易失性存储器可以包括RAM,例如,RAM可以用作外部高速缓存。通过说明而非限制,RAM有多种形式,例如同步RAM(SRAM)、动态RAM(DRAM)、同步DRAM(SDRAM)、双倍数据速率SDRAM(DDR

SDRAM)、增强型SDRAM(ESDRAM)、Synchlink DRAM(SLDRAm),直接Rambus RAM(DRRAM)、直接Rambus动态RAM(DRDRAM)和Rambus动态RAM(RDRAM)。另外,本文公开的系统或计算机实现的方法的存储器组件旨在包括但不限于包括这些以及任何其他合适类型的存储器。

[0088] 上面已经描述的内容仅包括系统和计算机实现的方法的示例。当然,出于描述本公开的目的,不可能描述组件或计算机实现的方法的每种可能的组合,但是本领域的普通技术人员可以认识到,本公开的许多进一步的组合和替换是可能的。此外,在说明书、权利要求书、附录和附图中的“包括”、“具有”等在使用时的解释旨在类似于权利要求中的术语“包括”。

[0089] 以上已经描述了各实施例,上述说明是示例性的,并非穷尽性的,并且也不限于所披露的各实施例。在不偏离所说明的各实施例的范围和精神的情况下,对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。本文中所用术语的选择,旨在最好地解释各实施例的原理、实际应用或对市场中的技术的技术改进,或者使本技术领域的其它普通技术人员能理解本文披露的各实施例。

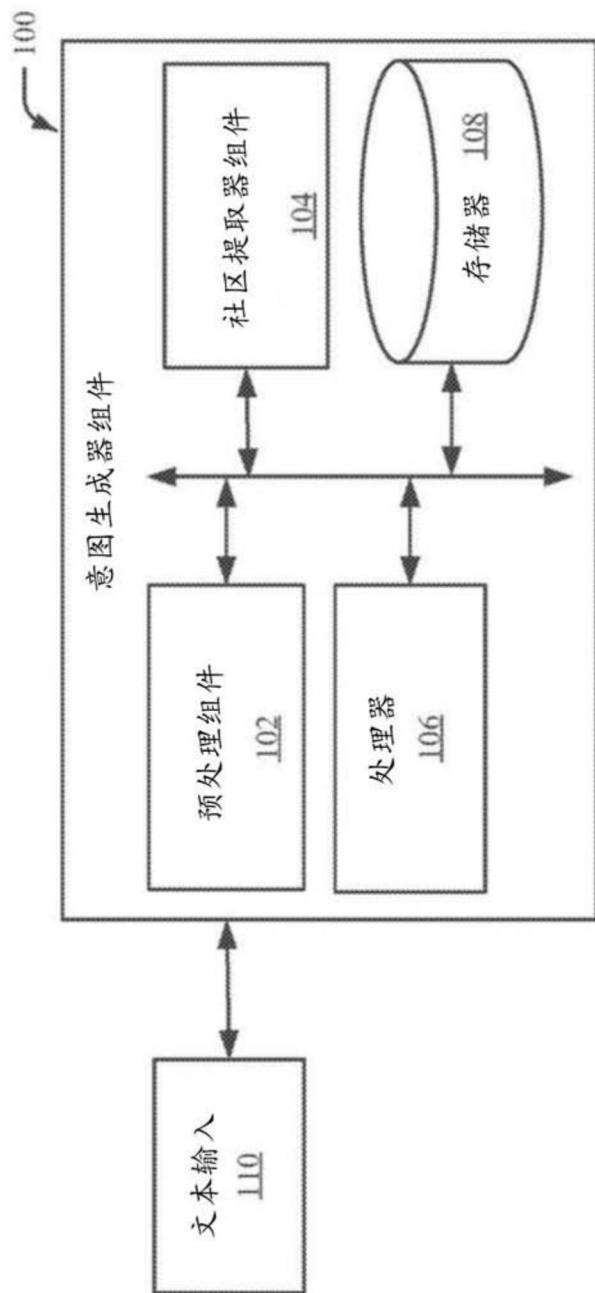


图1

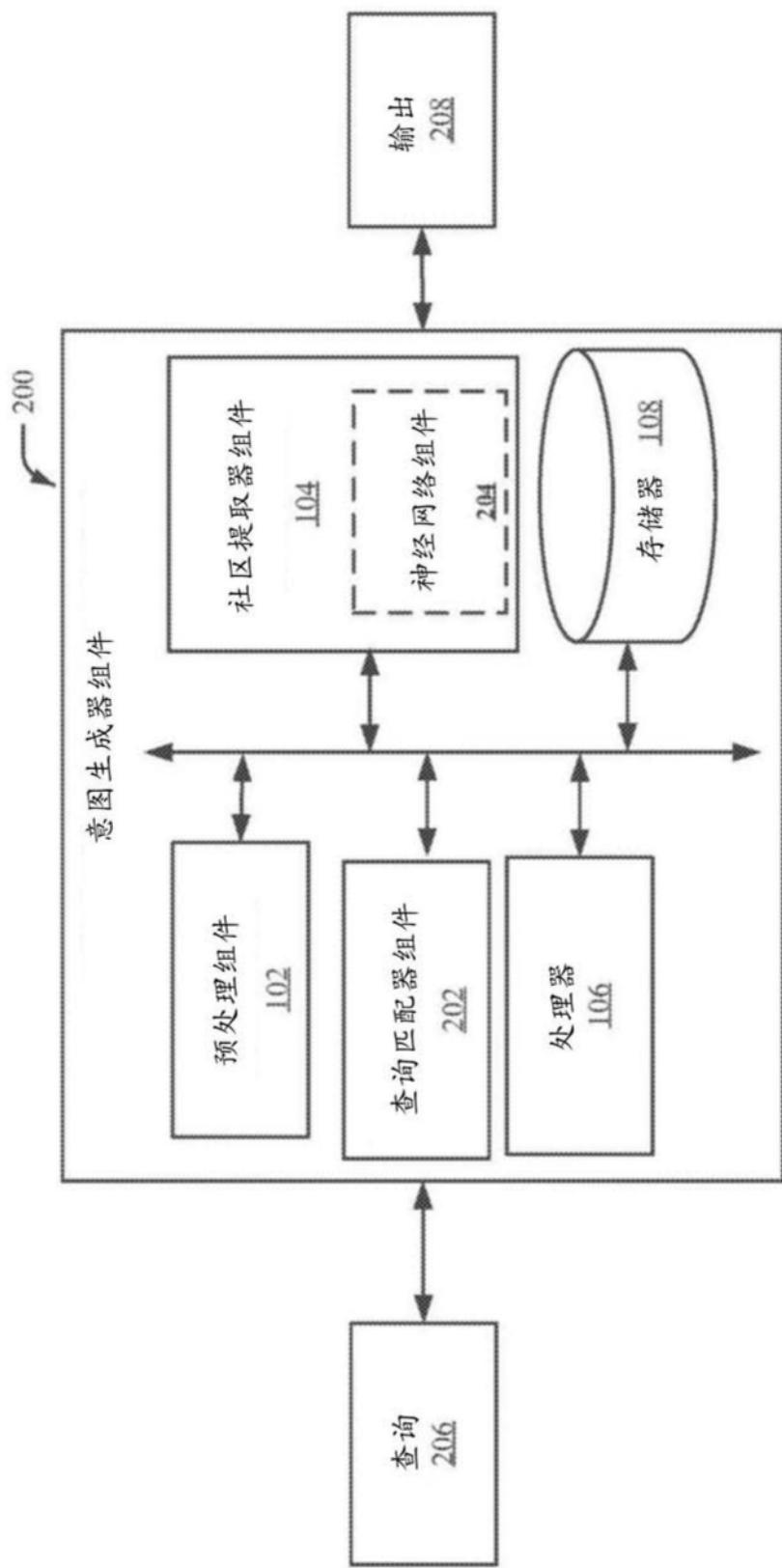


图2

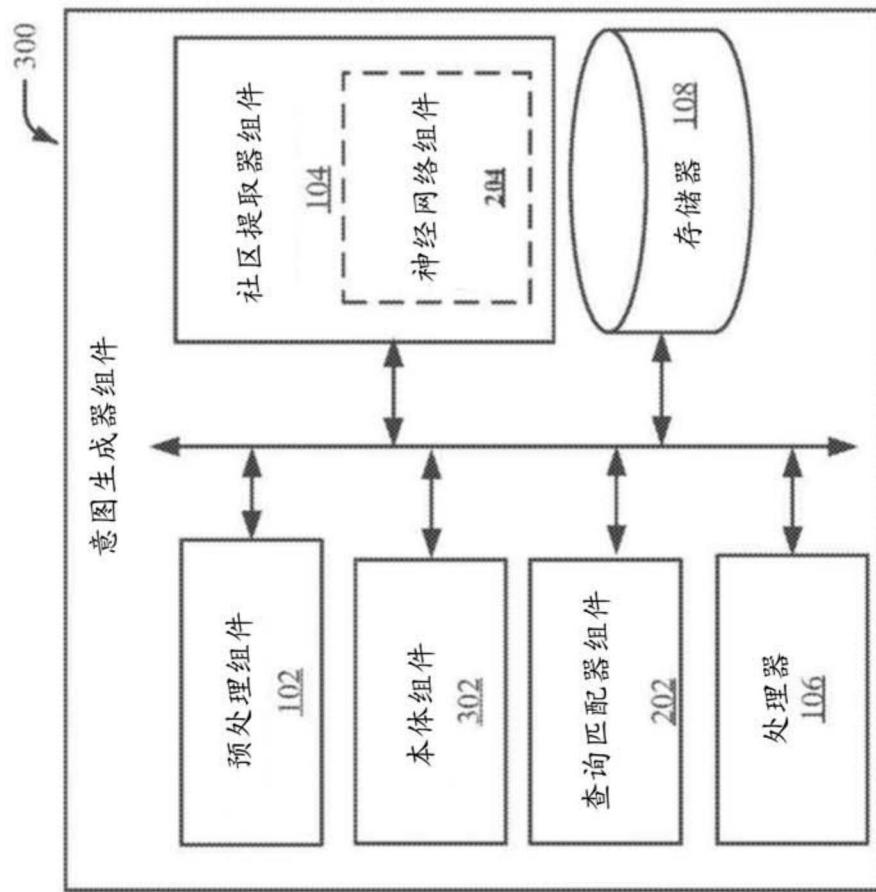


图3

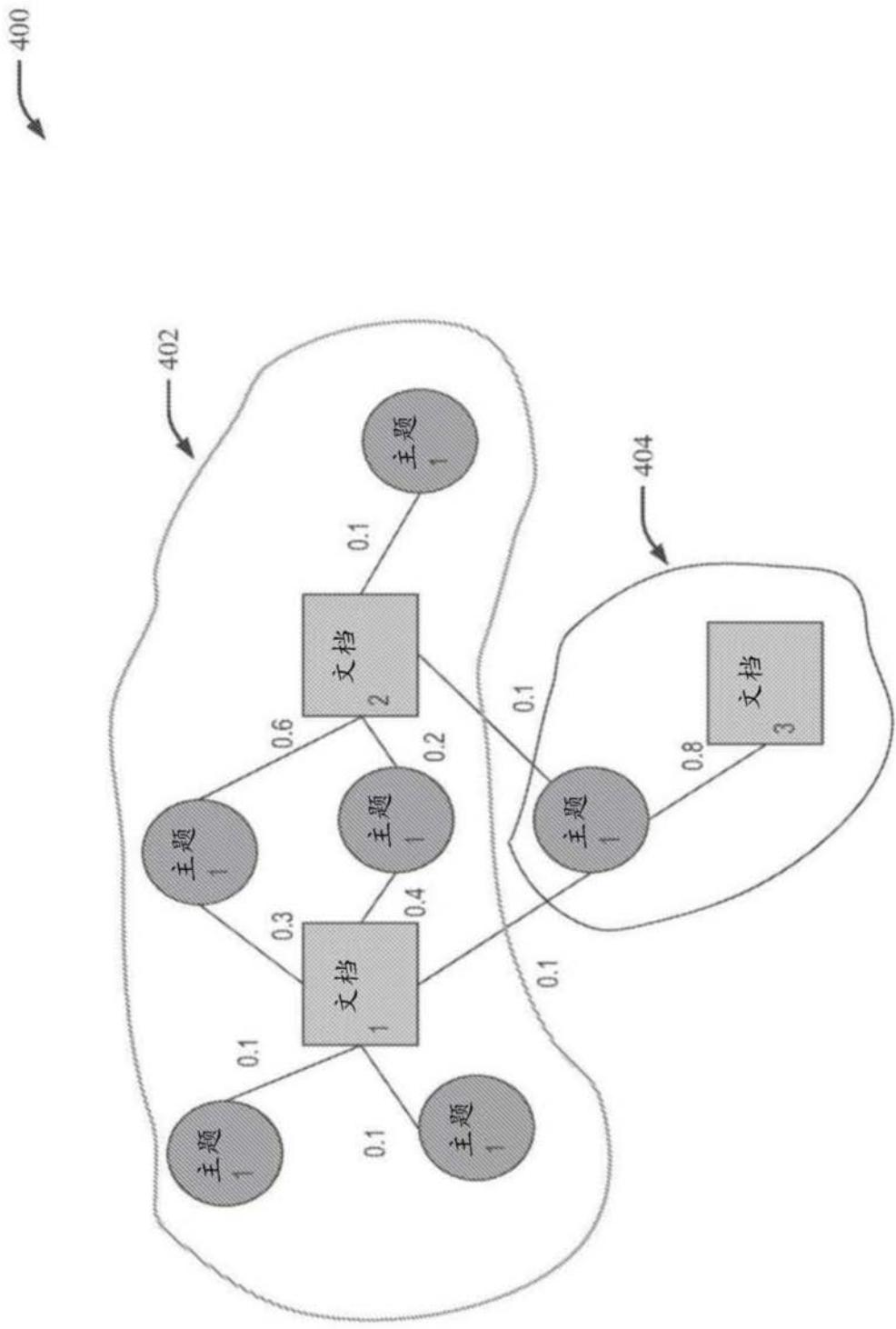


图4

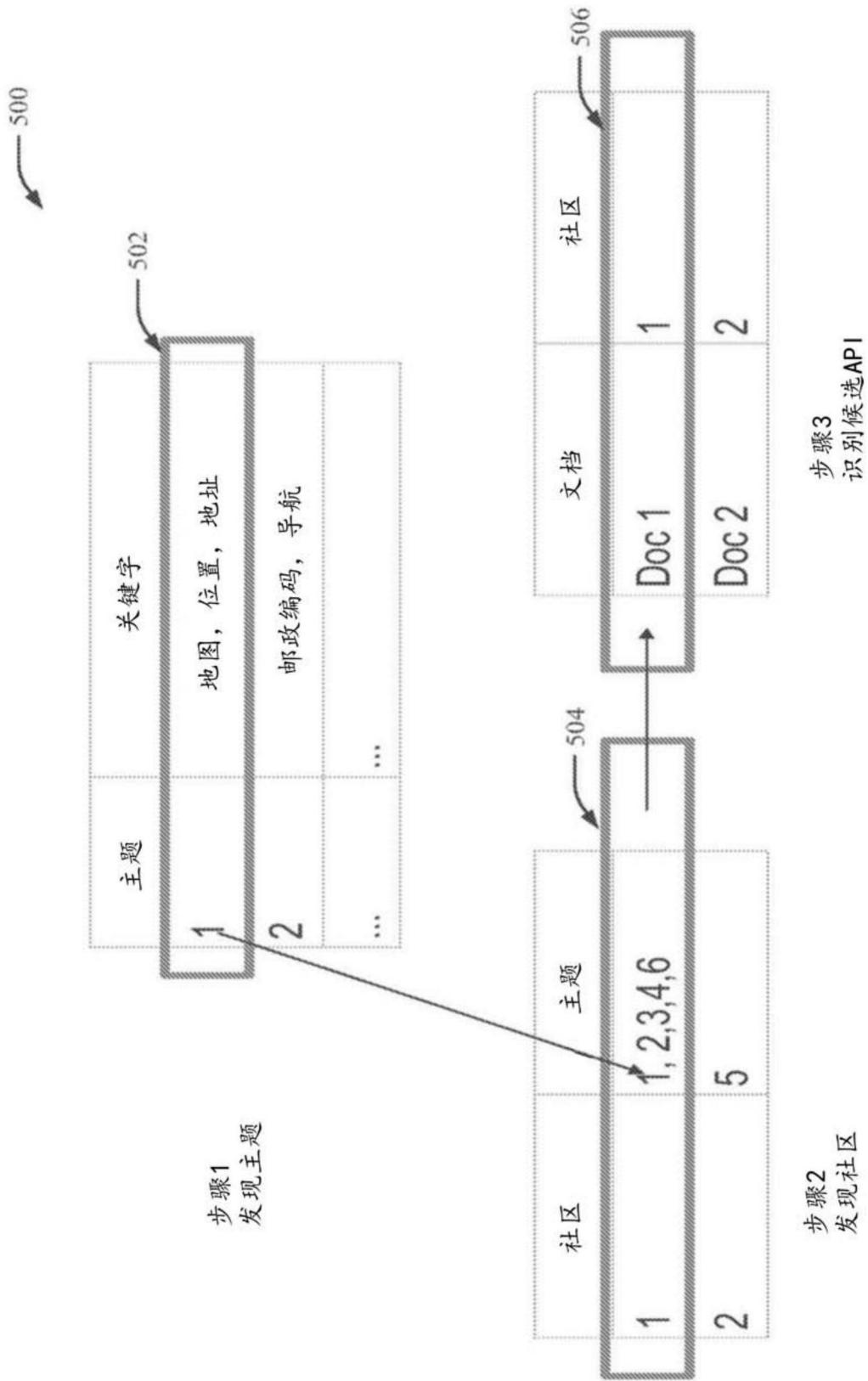


图5

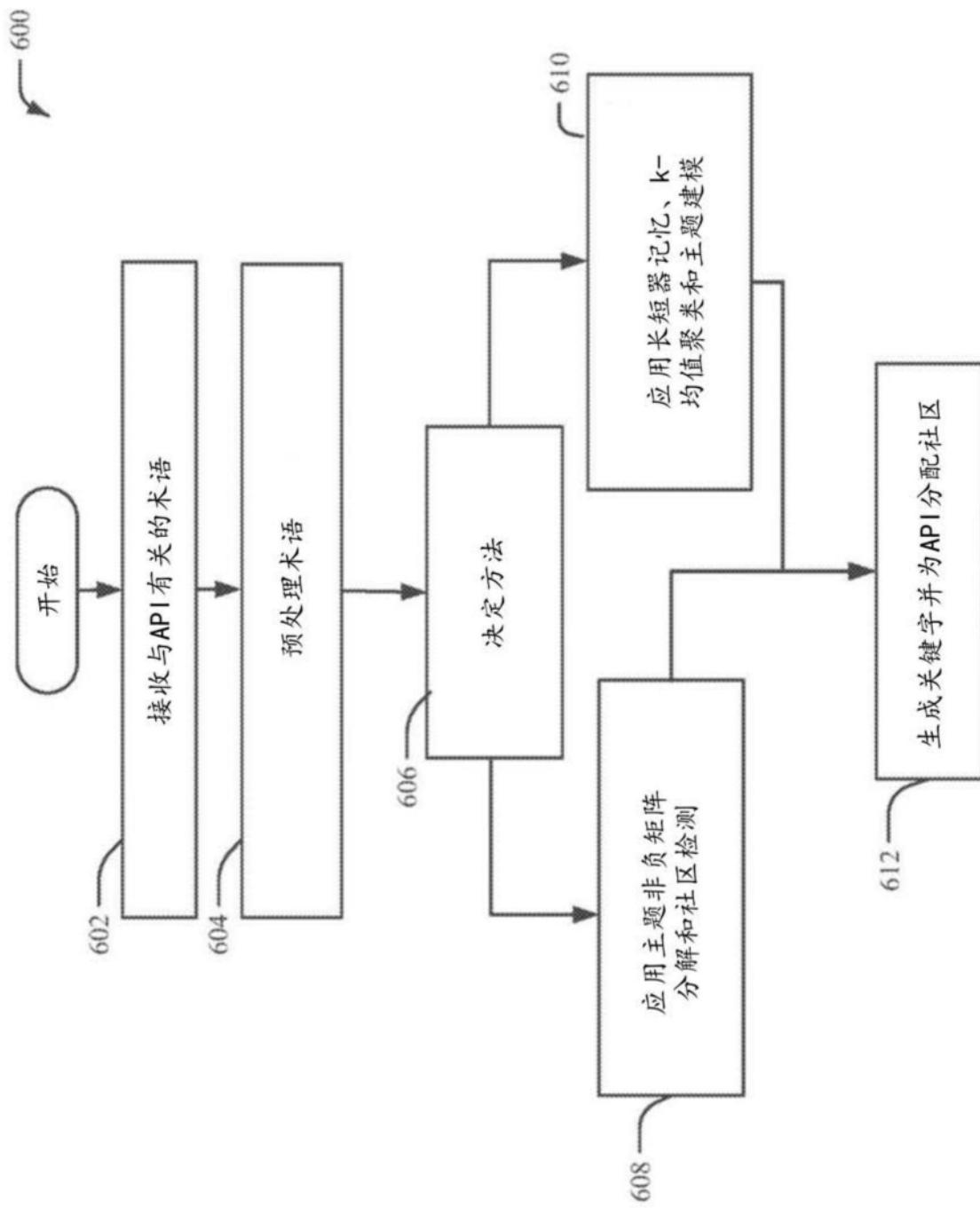


图6

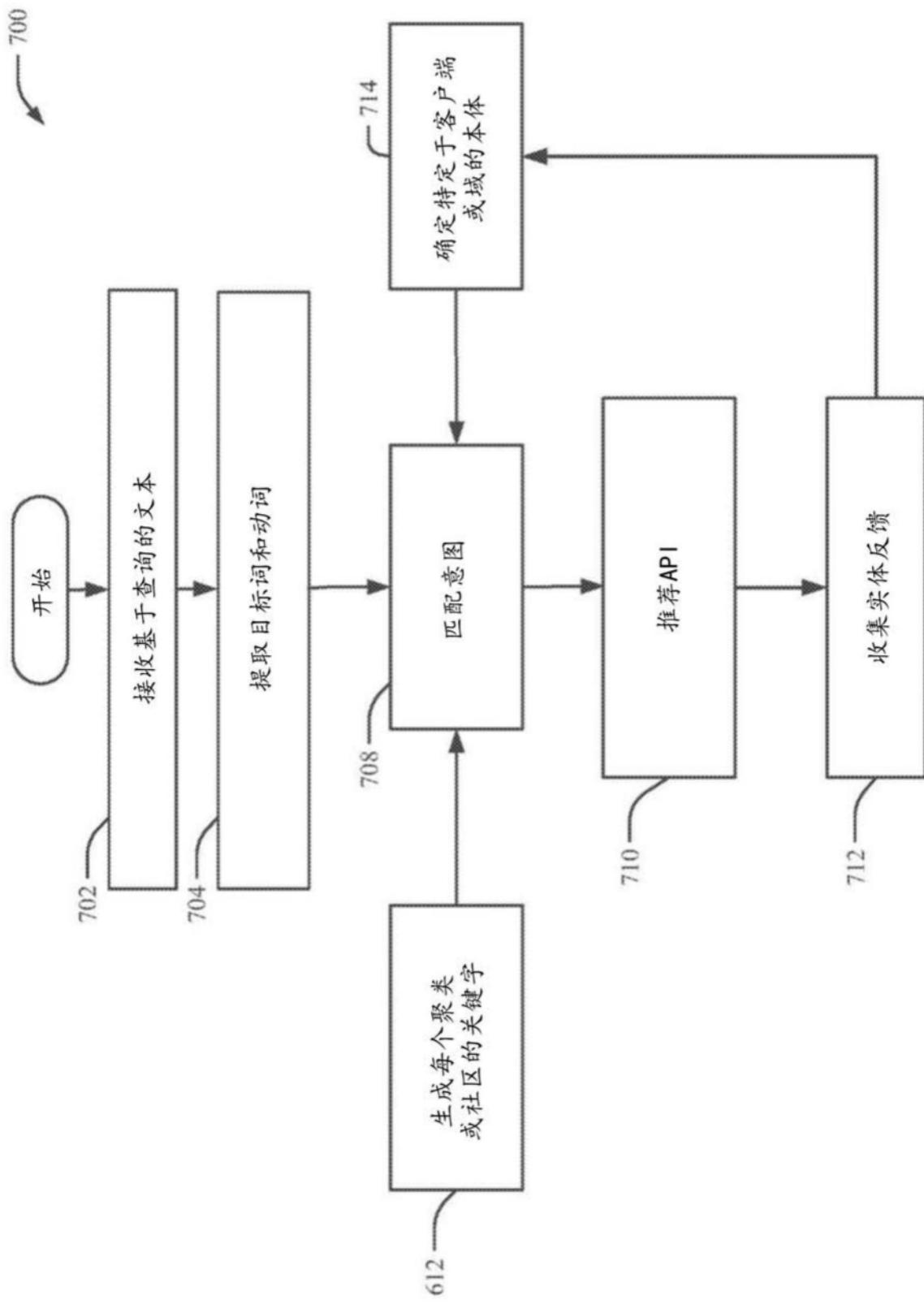


图7

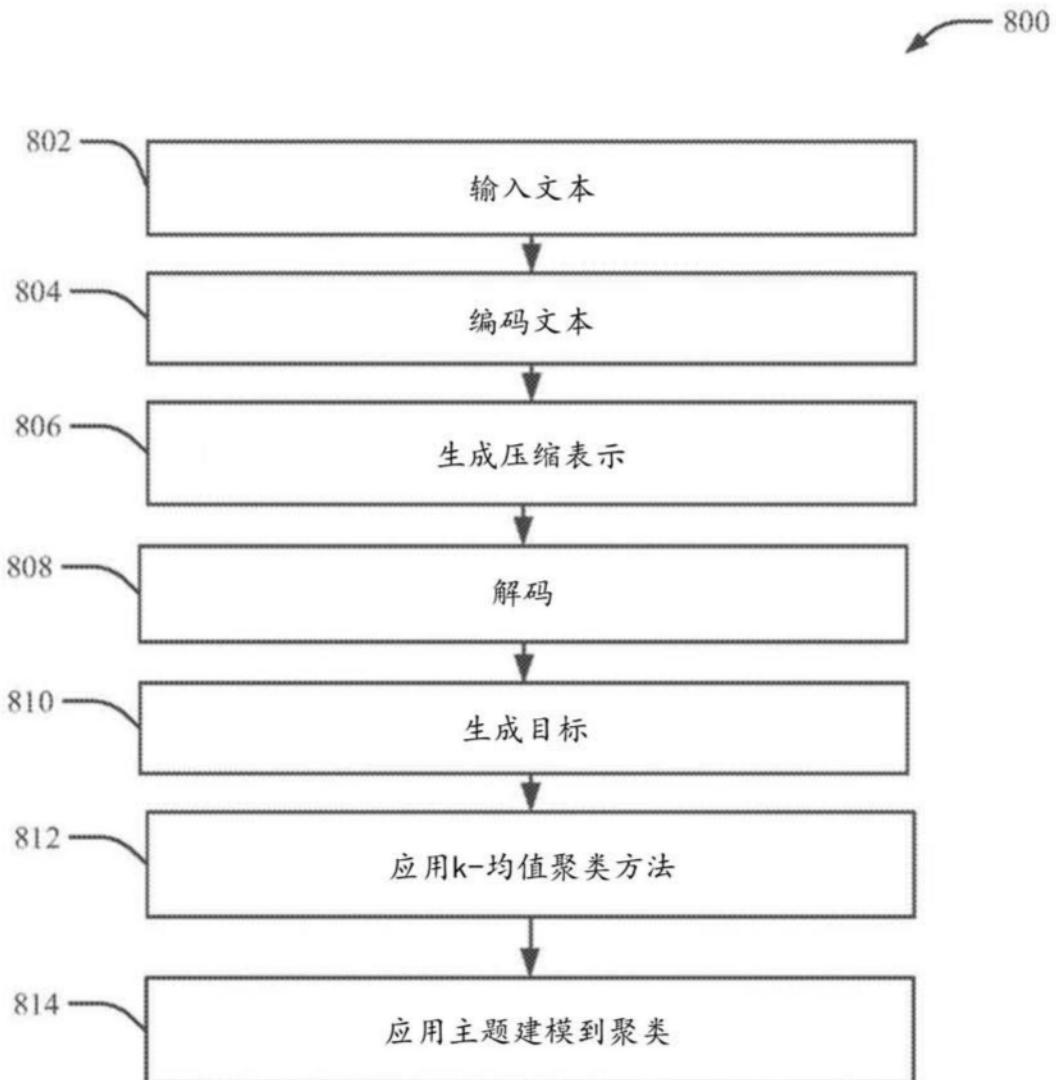


图8

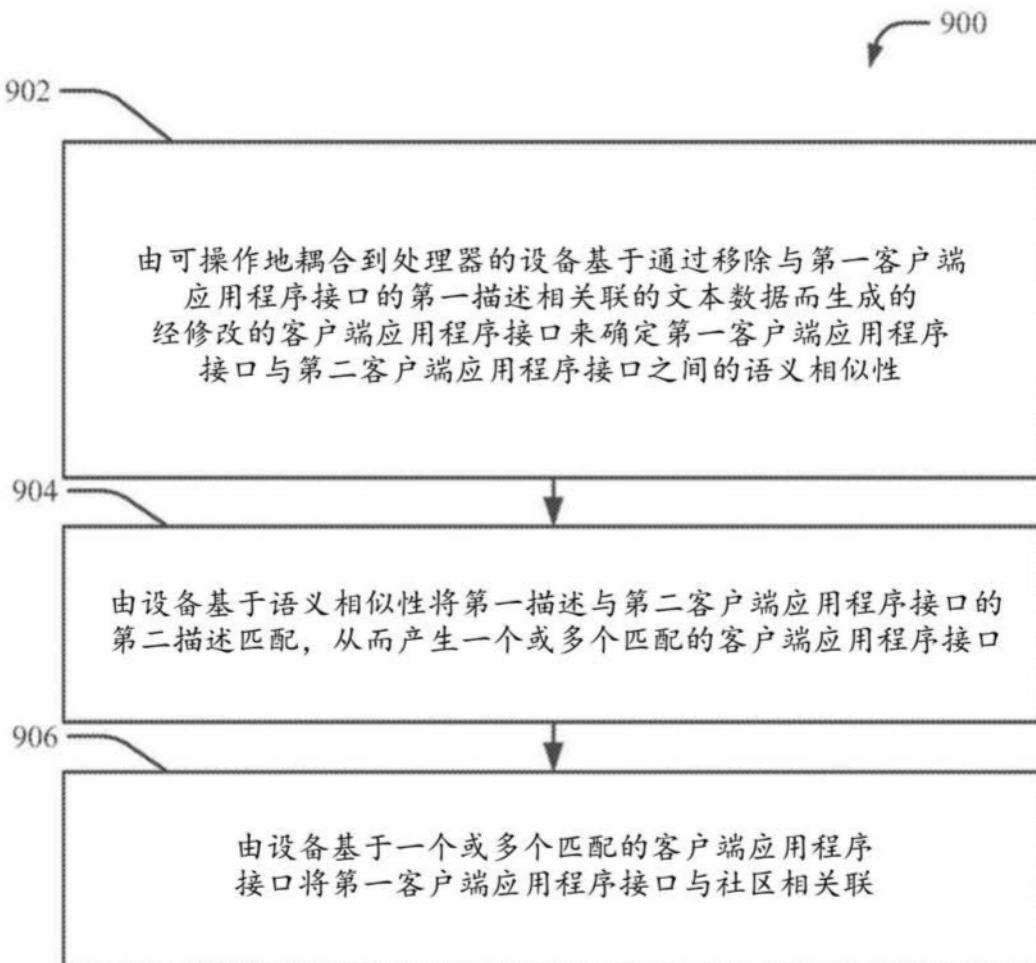


图9

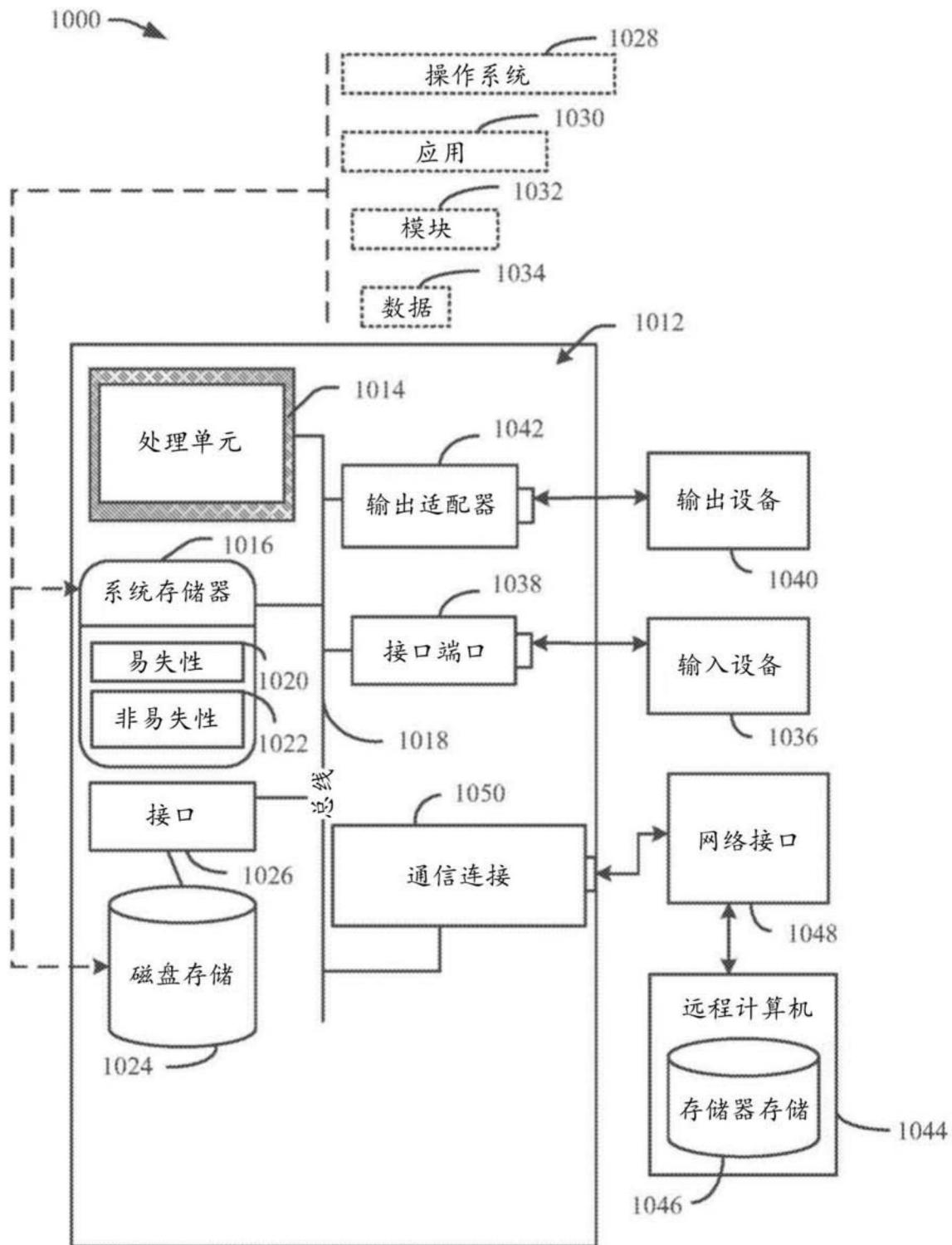


图10