



(12)发明专利申请

(10)申请公布号 CN 110442332 A

(43)申请公布日 2019. 11. 12

(21)申请号 201910600648.5

(22)申请日 2019.07.04

(71)申请人 深圳壹账通智能科技有限公司
地址 518000 广东省深圳市前海深港合作区前湾一路1号A栋201室(入驻深圳市前海商务秘书有限公司)

(72)发明人 刘利

(74)专利代理机构 深圳市明日今典知识产权代理有限公司(普通合伙) 44343
代理人 王杰辉

(51)Int.Cl.
G06F 8/33(2018.01)
G06F 17/22(2006.01)
G06F 16/33(2019.01)
G06F 16/903(2019.01)

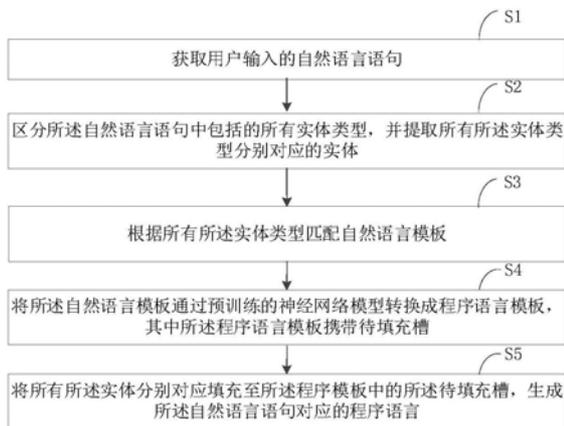
权利要求书2页 说明书14页 附图1页

(54)发明名称

自然语言转换成程序语言的方法、装置及计算机设备

(57)摘要

本申请揭示了自然语言转换成程序语言的方法、装置及计算机设备,其中,自然语言转换成程序语言的方法,包括:获取用户输入的自然语言语句;区分所述自然语言语句中包括的所有实体类型,并提取所有所述实体类型分别对应的实体;根据所有所述实体类型匹配自然语言模板;将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板,其中所述程序语言模板携带待填充槽;将所有所述实体分别对应填充至所述程序模板中的所述待填充槽,生成所述自然语言语句对应的程序语言。通过训练设计使填充程序模板中预留了待填充槽,用于将自然语言语句中的实体填充至程序模板中,形成自然语言语句对应的程序语言。



1. 一种自然语言转换成程序语言的方法,其特征在于,包括:

获取用户输入的自然语言语句;

区分所述自然语言语句中包括的所有实体类型,并提取所有所述实体类型分别对应的实体;

根据所有所述实体类型匹配自然语言模板;

将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板,其中,所述程序语言模板携带待填充槽;

将所有所述实体分别对应填充至所述程序模板中的所述待填充槽,生成所述自然语言语句对应的程序语言。

2. 根据权利要求1所述的自然语言转换成程序语言的方法,其特征在于,所述将所有所述实体分别对应填充至所述程序模板中的所述待填充槽,生成所述自然语言语句对应的程序语言的步骤,包括:

根据预设匹配计算公式确定与第一槽匹配的第一实体,其中,所述第一实体包含于所有所述实体中,所述第一槽包含于所有所述待填充的槽;

若是,则将所述第一实体填充至所述第一槽中;

按照所述第一槽的填充过程完成所述程序模板中的所有所述待填充槽的填充,生成所述程序语言。

3. 根据权利要求2所述的自然语言转换成程序语言的方法,其特征在于,所述根据预设匹配计算公式确定与第一槽匹配的第一实体的步骤,包括:

通过计算公式 $Y(i, j) = \sum_{(c, d) \in NN(i, j, k)} d_{(i, j), (c, d)} \cdot V(c, d)$, 分别计算所有所述实体与所述

第一槽之间的匹配分数值,其中 $d_{(i, j), (c, d)} = \cos((h_i, h_j^{\sim}), (h_c, h_d^{\sim}))$,

$V(c, d) = \begin{cases} 1, & \text{if } (e_c, s_d) \\ 0, & \text{otherwise} \end{cases}$, h_i 表示所有所述实体分别对应的向量, h_j^{\sim} 表示所述第一槽对应的向量, (e_c, s_d) 为已标注匹配状态的第二实体和第二槽, h_c 表示所述第二实体 e_c 对应的向量, h_d^{\sim} 表示所述第二槽 s_d 对应的向量;

确定所述匹配分数值最大时,与所述第一槽对应的指定实体;

将所述指定实体确定为所述第一实体。

4. 根据权利要求1所述的自然语言转换成程序语言的方法,其特征在于,所述神经网络模型包括编码-解码模型,所述将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板的步骤,包括:

将所述自然语言模板输入预训练的解码编码模型的隐藏层中,进行拟合得到拟合态自然语言模板并输出,其中,所述解码编码模型的隐藏层包含第一指定数量的形态,所述第一指定数量与所述自然语言模板中包含的实体类型的数量相同;

将所述拟合态自然语言模板输入预训练的解码模型的隐藏层,通过迭代输入计算转换为已排序的所述程序语言模板的列表并输出,其中所述程序语言模板按照预设格式保留所述待填充槽。

5. 根据权利要求4所述的自然语言转换成程序语言的方法,其特征在于,所述将所述拟合态自然语言模板输入预训练的解码模型的隐藏层,通过迭代输入计算转换为已排序的所

述程序语言模板的列表并输出的步骤,包括:

将所述拟合态自然语言模板作为第一输入数据,输入到解码模型的隐藏层的第一梯度层,确定权重系数最高的向量对应的第一实体类型并输出;

将所述第一实体类型与所述第一梯度层的计算数据作为第二输入数据,输入到解码模型的隐藏层的第二梯度层,确定权重系数次高的向量对应的第二实体类型并输出,其中所述第一梯度层和第二梯度层为所述解码模型的隐藏层中相邻连接的迭代层;

按照由所述第一输入数据到所述第二输入数据的迭代过程,依次迭代输入至所述解码模型的隐藏层,至确定出所述自然语言语句的所有主干词语,其中各所述主干词语分别根据各自对应的权重系数的排序确定,并根据权重系数的排序,形成排序的所述程序语言模板的列表。

6. 根据权利要求1所述的自然语言转换成程序语言的方法,其特征在于,所述区分所述自然语言语句中包括的所有实体类型,并提取所有所述实体类型分别对应的实体的步骤,包括:

调用根据所述程序模板的构成要素预配置的正则表达式,其中所述正则表达式的类型至少包括两种;

根据第一正则表达式判断所述自然语言语句中,是否包含与所述第一正则表达式适配的指定实体类型,其中,所述第一正则表达式包含于所有所述正则表达式中;

若是,则通过所述第一正则表达式提取所述指定实体类型对应的实体。

7. 根据权利要求1所述的自然语言转换成程序语言的方法,其特征在于,所述实体类型包括模式类实体和数量类实体,所述根据所有所述实体类型匹配自然语言模板的步骤,包括:

分别获取所述模式类实体对应的实体和实体数量,以及所述数量类实体对应的实体和实体数量;

根据所述模式类实体对应的实体和实体数量,以及所述数量类实体对应的实体和实体数量,按照预设次序排列所述模式类实体对应的实体和以及所述数量类实体对应的实体,形成所述自然语言模板。

8. 一种自然语言转换成程序语言的装置,其特征在于,包括:

获取模块,用于获取用户输入的自然语言语句;

区分模块,用于区分所述自然语言语句中包括的所有实体类型,并提取所有所述实体类型分别对应的实体;

匹配模块,用于根据所有所述实体类型匹配自然语言模板;

转换模块,用于将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板,其中,所述程序语言模板携带待填充槽;

填充模块,用于将所有所述实体分别对应填充至所述程序模板中的所述待填充槽,生成所述自然语言语句对应的程序语言。

9. 一种计算机设备,包括存储器和处理器,所述存储器存储有计算机程序,其特征在于,所述处理器执行所述计算机程序时实现权利要求1至7中任一项所述方法的步骤。

10. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现权利要求1至7中任一项所述的方法的步骤。

自然语言转换成程序语言的方法、装置及计算机设备

技术领域

[0001] 本申请涉及到计算机领域,特别是涉及到自然语言转换成程序语言的方法、装置及计算机设备。

背景技术

[0002] 现有技术中,没有程序语言和自然语言语句的对译工具,通常情况下程序员在编程过程中可能会遇到可以用自然语言描述他构想好的实现方案,但是在将其翻译成代码过程中遇到技术困难;而且现有的互联网资源,比如问答类型的网站,将程序员提交的编程问题制成程序表格,当设计新任务或复合任务,需要结合历史的编程内容时,无法通过自然语言语句实现对相应编程内容的有效的查找和调用。

发明内容

[0003] 本申请的主要目的为提供自然语言转换成程序语言的方法,旨在解决无法通过自然语言语句实现对相应编程内容的有效的查找和调用的技术问题。

[0004] 本申请提出一种自然语言转换成程序语言的方法,包括:

[0005] 获取用户输入的自然语言语句;

[0006] 区分所述自然语言语句中包括的所有实体类型,并提取所有所述实体类型分别对应的实体;

[0007] 根据所有所述实体类型匹配自然语言模板;

[0008] 将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板,其中所述程序语言模板携带待填充槽;

[0009] 将所有所述实体分别对应填充至所述程序模板中的所述待填充槽,生成所述自然语言语句对应的程序语言。

[0010] 优选地,所述将所有所述实体分别对应填充至所述程序模板中的所述待填充槽,生成所述自然语言语句对应的程序语言的步骤,包括:

[0011] 根据预设匹配计算公式确定与第一槽匹配的第一实体,其中,所述第一实体包含于所有所述实体中,所述第一槽包含于所有所述待填充的槽;

[0012] 若是,则将所述第一实体填充至所述第一槽中;

[0013] 按照所述第一槽的填充过程完成所述程序模板中的所有所述待填充槽的填充,生成所述程序语言。

[0014] 优选地,所述根据预设匹配计算公式确定与第一槽匹配的第一实体的步骤,包括:

[0015] 通过计算公式 $Y(i, j) = \sum_{(c, d) \in NN(i, j, k)} d_{(i, j), (c, d)} \cdot V(c, d)$, 分别计算所有所述实体与

所述第一槽之间的匹配分数值,其中 $d_{(i, j), (c, d)} = \cos((h_i, h_j^{\sim}), (h_c, h_d^{\sim}))$,

$V(c, d) = \begin{cases} 1, & \text{if}(e_c, s_d) \\ 0, & \text{otherwise} \end{cases}$, h_i 表示所有所述实体分别对应的向量, h_j^{\sim} 表示所述第一槽对应

的向量, (e_c, s_d) 为已标注匹配状态的第二实体和第二槽, h_c 表示所述第二实体 e_c 对应的向量, h_d 表示所述第二槽 s_d 对应的向量;

[0016] 确定所述匹配分数值最大时, 与所述第一槽对应的指定实体;

[0017] 将所述指定实体确定为所述第一实体。

[0018] 优选地, 所述神经网络模型包括编码-解码模型, 所述将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板的步骤, 包括:

[0019] 将所述自然语言模板输入预训练的解码编码模型的隐藏层中, 进行拟合得到拟合态自然语言模板并输出, 其中, 所述解码编码模型的隐藏层包含第一指定数量的形态, 所述第一指定数量与所述自然语言模板中包含的实体类型的数量相同;

[0020] 将所述拟合态自然语言模板输入预训练的解码模型的隐藏层, 通过迭代输入计算转换为已排序的所述程序语言模板的列表并输出, 其中所述程序语言模板按照预设格式保留所述待填充槽。

[0021] 优选地, 所述将所述拟合态自然语言模板输入预训练的解码模型的隐藏层, 通过迭代输入计算转换为已排序的所述程序语言模板的列表并输出的步骤, 包括:

[0022] 将所述拟合态自然语言模板作为第一输入数据, 输入到解码模型的隐藏层的第一梯度层, 确定权重系数最高的向量对应的第一实体类型并输出;

[0023] 将所述第一实体类型与所述第一梯度层的计算数据作为第二输入数据, 输入到解码模型的隐藏层的第二梯度层, 确定权重系数次高的向量对应的第二实体类型并输出, 其中所述第一梯度层和第二梯度层为所述解码模型的隐藏层中相邻连接的迭代层;

[0024] 按照由所述第一输入数据到所述第二输入数据的迭代过程, 依次迭代输入至所述解码模型的隐藏层, 至确定出所述自然语言语句的所有主干词语, 其中各所述主干词语分别根据各自对应的权重系数的排序确定, 并根据权重系数的排序, 形成排序的所述程序语言模板的列表。

[0025] 优选地, 所述区分所述自然语言语句中包括的所有实体类型, 并提取所有所述实体类型分别对应的实体的步骤, 包括:

[0026] 调用根据所述程序模板的构成要素预配置的正则表达式, 其中所述正则表达式的类型至少包括两种;

[0027] 根据第一正则表达式判断所述自然语言语句中, 是否包含与所述第一正则表达式适配的指定实体类型, 其中, 所述第一正则表达式包含于所有所述正则表达式中;

[0028] 若是, 则通过所述第一正则表达式提取所述指定实体类型对应的实体。

[0029] 优选地, 所述实体类型包括模式类实体和数量类实体, 所述根据所有所述实体类型匹配自然语言模板的步骤, 包括:

[0030] 分别获取所述模式类实体对应的实体和实体数量, 以及所述数量类实体对应的实体和实体数量;

[0031] 根据所述模式类实体对应的实体和实体数量, 以及所述数量类实体对应的实体和实体数量, 按照预设次序排列所述模式类实体对应的实体和以及所述数量类实体对应的实体, 形成所述自然语言模板。

[0032] 本申请还提供了一种自然语言转换成程序语言的装置, 包括:

[0033] 获取模块, 用于获取用户输入的自然语言语句;

[0034] 区分模块,用于区分所述自然语言语句中包括的所有实体类型,并提取所有所述实体类型分别对应的实体;

[0035] 匹配模块,用于根据所有所述实体类型匹配自然语言模板;

[0036] 转换模块,用于将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板,其中,所述程序语言模板携带待填充槽;

[0037] 填充模块,用于将所有所述实体分别对应填充至所述程序模板中的所述待填充槽,生成所述自然语言语句对应的程序语言。

[0038] 本申请还提供了一种计算机设备,包括存储器和处理器,所述存储器存储有计算机程序,所述处理器执行所述计算机程序时实现上述方法的步骤。

[0039] 本申请还提供了一种计算机可读存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现上述的方法的步骤。

[0040] 本申请通过将自然语言语句中的实体类型提取出来,并根据提取的实体类型生成自然语言模板,然后将自然语言模板输入到训练好的神经网络中,将自然语言模板通过拟合计算以及迭代计算,转换成程序模板,并通过训练设计使填充程序模板中预留了待填充槽,用于将自然语言语句中的实体填充至程序模板中,形成自然语言语句对应的程序语言。通过将自然语言语句直接转换为程序语言,实现能够让编程人员使用自然语言来表达预期的操作,方便程序员审查和使用。通过局部匹配函数改进了k-近邻算法来完成,解决了现有k-近邻算法中的欧式距离等不适用于本申请中的槽和实体的匹配的问题,通过将距离最近的实体和槽组成的数据对,拼接为一个长向量,通过计算两个长向量之间的相似度,进而得到待匹配的槽和实体之间的匹配度。通过上述方式实现槽与实体的准确对齐,提高从自然语言语句到程序语言的准确对应性。通过匹配不同的正则表达式来识别不同的实体类型,并提取相应的实体,方法简便,计算量小且识别提取实体的准确度高。通过将自然语言语句依次输入到各正则表达式中,以判断自然语言语句中是否包括与当前正则表达式匹配的实体类型,若存在,则提取出该实体。

附图说明

[0041] 图1本申请一实施例的自然语言转换成程序语言的方法流程示意图;

[0042] 图2本申请一实施例的自然语言转换成程序语言的装置结构示意图;

[0043] 图3本申请一实施例的计算机设备内部结构示意图。

具体实施方式

[0044] 为了使本申请的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本申请进行进一步详细说明。应当理解,此处描述的具体实施例仅仅用以解释本申请,并不用于限定本申请。

[0045] 参照图1,本申请一实施例的自然语言转换成程序语言的方法,包括:

[0046] S1:获取用户输入的自然语言语句。

[0047] S2:区分所述自然语言语句中包括的所有实体类型,并提取所有所述实体类型分别对应的实体。

[0048] S3:根据所有所述实体类型匹配自然语言模板。

[0049] S4:将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板,其中,所述程序语言模板携带待填充槽。

[0050] S5:将所有所述实体分别对应填充至所述程序模板中的所述待填充槽,生成所述自然语言语句对应的程序语言。

[0051] 本实施例通过将自然语言语句中的实体类型提取出来,并根据提取的实体类型生成自然语言模板,然后将自然语言模板输入到训练好的神经网络中,将自然语言模板通过拟合计算以及迭代计算,转换成程序模板,并通过训练设计使填充程序模板中预留了待填充槽,用于将自然语言语句中的实体填充至程序模板中,形成自然语言语句对应的程序语言。上述实体类型为用于表示自然语言语句的主干词语对应的词语类型,比如包括文件名,路径名,绝对路径名、时间等,上述的实体包括文件名,路径名,绝对路径名、时间等分别对应的赋值。上述提取实体类型的过程,为识别语义确定主干词语的过程,即提取出的词保留了原自然语言语句的原始语义表达。上述自然语言模板为上述主干词语分别对应的实体类型,按照在原自然语言语句中的排列次序而形成的序列。然后将上述序列输入到训练好的神经网络中,通过编码和解码中的拟合和迭代计算,使自然语言模板逐渐转变为程序模板的序列,并预留了待填充槽。本实施例通过将自然语言语句直接转换为程序语言,实现能够让编程人员使用自然语言来表达预期的操作,方便程序员审查和使用。

[0052] 进一步地,所述将所有所述实体分别对应填充至所述程序模板中的所述待填充槽,生成所述自然语言语句对应的程序语言的步骤S5,包括:

[0053] S51:根据预设匹配计算公式确定与第一槽匹配的第一实体,其中,所述第一实体包含于所有所述实体中,所述第一槽包含于所有所述待填充的槽。

[0054] S52:若是,则将所述第一实体填充至所述第一槽中。

[0055] S53:按照所述第一槽的填充过程完成所述程序模板中的所有所述待填充槽的填充,生成所述程序语言。

[0056] 本实施例通过将实体和槽进行匹配配对,实现将实体准确填充于槽,解决了将正确的输入字符串与输出命令的参数值对齐。本实施例的预设匹配计算公式为通过局部匹配函数改进了k-近邻算法来完成的,由于现有k-近邻算法中的欧式距离等不适用于本申请中的槽和实体的匹配,本申请中的槽和实体无具体对应的向量,本实施例通过将距离最近的实体和槽组成的数据对,拼接为一个长向量,通过计算两个长向量之间的相似度,进而得到待匹配的槽和实体之间的匹配度。通过上述方式实现槽与实体的准确对齐,提高从自然语言语句到程序语言的准确对应性。

[0057] 进一步地,所述根据预设匹配计算公式确定与第一槽匹配的第一实体的步骤S51,包括:

[0058] S511:通过计算公式 $Y(i, j) = \sum_{(c, d) \in NN(i, j, k)} d_{(i, j), (c, d)} \cdot V(c, d)$, 分别计算所有所述实

体与所述第一槽之间的匹配分数值,其中 $d_{(i, j), (c, d)} = \cos((h_i, h_j), (h_c, h_d))$,

$V(c, d) = \begin{cases} 1, & \text{if}(e_c, s_d) \\ 0, & \text{otherwise} \end{cases}$, h_i 表示所有所述实体分别对应的向量, h_j 表示所述第一槽对应的

的向量, (e_c, s_d) 为已标注匹配状态的第二实体和第二槽, h_c 表示所述第二实体 e_c 对应的向量, h_d 表示所述第二槽 s_d 对应的向量。

[0059] S512:确定所述匹配分数值最大时,与所述第一槽对应的指定实体。

[0060] S513:将所述指定实体确定为所述第一实体。

[0061] 本实施例中,通过局部匹配函数改进了k-近邻算法,根据k-近邻算法通过计算在相邻区域内出现概率,来判断相似性的原理,改进成通过选择距离待匹配实体和槽数据对最近的已标记匹配数据的实体和槽数据对,并计算上述两对实体和槽数据对的匹配分数值,来克服现有实体和槽之间无法直接实现准确映射向量的问题,且通过改进的k-近邻算法,减少了计算量且提高了实体和槽的匹配准确度。举例地,上述第二实体 e_c 和第二槽 s_d ,分别在编码模型(Encoder model)和解码模型(Decoder model)里对应一个隐藏层变量 h_c, h_d ,通过将变量拼接,形成长向量 (h_c, h_d) ,组成了参照实体和槽数据对,且标识了匹配状态,匹配标识为1,比匹配标识为0。通过已确定的待填充槽,即第一槽,分别匹配所有的实体,组成多个包括第一槽的实体和槽数据对,同样 h_i 表示所有实体分别在Encoder模型里对应的隐藏层变量, h_j 表示所述第一槽Decoder模型里对应的隐藏层变量,通过拼接形成长向量 (h_i, h_j) ,通过筛选距离待分析的第一槽的实体和槽数据对,距离值最近的已标注匹配状态的第二实体和第二槽,并计算与已标注匹配状态的第二实体和第二槽的匹配分数值,匹配分数值越大,则表明待分析的第一槽的实体和槽数据对,与已标注匹配状态的第二实体和第二槽越相似。比如已标注匹配状态的第二实体和第二槽标注为匹配,则跟它相似的待分析的第一槽的实体和槽数据对也推定为匹配,实现实体和槽的准确对应。

[0062] 进一步地,所述神经网络模型包括编码-解码模型,所述将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板的步骤S4,包括:

[0063] S41:将所述自然语言模板输入预训练的解码编码模型的隐藏层中,进行拟合得到拟合态自然语言模板并输出,其中,所述解码编码模型的隐藏层包含第一指定数量的形态,所述第一指定数量与所述自然语言模板中包含的实体类型的数量相同。

[0064] S42:将所述拟合态自然语言模板输入预训练的解码模型的隐藏层,通过迭代输入计算转换为已排序的所述程序语言模板的列表并输出,其中所述程序语言模板按照预设格式保留所述待填充槽。

[0065] 本实施例通过Encoder的隐藏层实现对自然语言模板的降维拟合,降低处理过程中的复杂度,且提高了对自然语言模板的提取特征的正确识别率。本实施例中Encoder包括4层隐藏层对应7个拟合形态依次为H1至H7,将每层LSTM的步长设置为20;在MSCOCO数据集上,其隐层输出设置为1000,在Flickr30K数据集上,LSTM隐层输出为512,在Flickr8K上,隐层输出设置为256,以提高降维效果。本实施例的Decoder的隐藏层通过接收Encoder输出的最后一个隐藏层拟合形态作为输入,经过多次迭代输入输出,得到已排序的所述程序语言模板的列表,并按照预设模式保留了待填充槽。上述的Encoder的隐藏层和Decoder的隐藏层相连,且均通过已标记的训练样本训练出最佳参量,使Encoder-Decoder模型输出的程序模板无限接近预设模式。

[0066] 进一步地,所述将所述拟合态自然语言模板输入预训练的解码模型的隐藏层,通过迭代输入计算转换为已排序的所述程序语言模板的列表并输出的步骤S42,包括:

[0067] S421:将所述拟合态自然语言模板作为第一输入数据,输入到解码模型的隐藏层的第一梯度层,确定权重系数最高的向量对应的第一实体类型并输出。

[0068] S422:将所述第一实体类型与所述第一梯度层的计算数据作为第二输入数据,输

入到解码模型的隐藏层的第二梯度层,确定权重系数次高的向量对应的第二实体类型并输出,其中所述第一梯度层和第二梯度层为所述解码模型的隐藏层中相邻连接的迭代层。

[0069] S423:按照由所述第一输入数据到所述第二输入数据的迭代过程,依次迭代输入至所述解码模型的隐藏层,至确定出所述自然语言语句的所有主干词语,其中各所述主干词语分别根据各自对应的权重系数的排序确定,并根据权重系数的排序,形成排序的所述程序语言模板的列表。

[0070] 本实施例通过多个相互连接的梯度层,进行迭代输入输出,比如包括H1`至H5`,将Encoder输出的最后一个隐藏层拟合形态,即拟合态自然语言模板作为第一输入数据,输入到Decoder的隐藏层的H1`,输出对自然语言语句的原语义最重要的第一实体类型Y1`,然后将Y1`以及计算提取Y1`的过程数据同时作为H2`的输入,输出对自然语言语句的原语义次重要的第二实体类型Y2`,然后将Y2`以及计算提取Y2`的过程数据同时作为H3`的输入,依次迭代进行计算提取,直至输出包括预设数量的所有实体类型的所述程序语言模板的列表。举例地,用户输入“查找15天前的日志文件”,对应的自然语言模板为:find all[file name]files older than[Timespan],经过Encoder拟合输出后,经过Decoder的隐藏层的迭代输入输出计算,得到的排序的所述程序语言模板的列表为:Y1:find[path]-name[regex]-mtime[+Timespan];Y2:find[path]-type f-name[regex]-mtime[+Timespan];Y3:find[path]-type f-perm[permission];Y4:find[path]-name[regex]-mtime[+Timespan]xargs ls;Y5:....

[0071] 进一步地,所述区分所述自然语言语句中包括的所有实体类型,并提取所有所述实体类型分别对应的实体的步骤S2,包括:

[0072] S21:调用根据所述程序模板的构成要素预配置的正则表达式,其中所述正则表达式的类型至少包括两种。

[0073] S22:根据第一正则表达式判断所述自然语言语句中,是否包含与所述第一正则表达式适配的指定实体类型,其中,所述第一正则表达式包含于所有所述正则表达式中。

[0074] S23:若是,则通过所述第一正则表达式提取所述指定实体类型对应的实体。

[0075] 本实施例通过匹配不同的正则表达式来识别不同的实体类型,并提取相应的实体,方法简便,计算量小且识别提取实体的准确度高。通过将自然语言语句依次输入到各正则表达式中,以判断自然语言语句中是否包括与当前正则表达式匹配的实体类型,若存在,则提取出该实体。本实施例的多个正则表达式的排布可预先根据各实体类型的重要程度,依次降序排布,以便识别的实体类型按照重要次序形成排列。

[0076] 进一步地,所述实体类型包括模式类实体和数量类实体,所述根据所有所述实体类型匹配自然语言模板的步骤S3,包括:

[0077] S31:分别获取所述模式类实体对应的实体和实体数量,以及所述数量类实体对应的实体和实体数量。

[0078] S32:根据所述模式类实体对应的实体和实体数量,以及所述数量类实体对应的实体和实体数量,按照预设次序排列所述模式类实体对应的实体和以及所述数量类实体对应的实体,形成所述自然语言模板。

[0079] 本实施例的实体类型包括多个模式类实体和多个数量类实体,比如包括5个模式类实体,包括:“File:file name;Directory:directory name;Path:absolute path;Date/

Time:date and time expression;Regex:other pattern arguments”,以及3个数量类实体,包括“Number:number;Size:file size;Timespan:time duration”,以尽量全面覆盖自然语言语句的所有主干词语。通过确定不同实体类型,以及各不同实体类型对应自然语言语句的重要性排序,形成自然语言模板,同时确定并提取各不同实体类型分别对应的实体,作为待填充槽的赋值。比如,用户输入“查找15天前的日志文件,即find all log files older than 15days”,主干词语包括“file name:log;Timespan:15days”,对应的自然语言模板为:find all[file name]files older than[Timespan]。

[0080] 参照图2,本申请一实施例的自然语言转换成程序语言的装置,包括:

[0081] 获取模块1,用于获取用户输入的自然语言语句。

[0082] 区分模块2,用于区分所述自然语言语句中包括的所有实体类型,并提取所有所述实体类型分别对应的实体。

[0083] 匹配模块3,用于根据所有所述实体类型匹配自然语言模板。

[0084] 转换模块4,用于将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板,其中,所述程序语言模板携带待填充槽。

[0085] 填充模块5,用于将所有所述实体分别对应填充至所述程序模板中的所述待填充槽,生成所述自然语言语句对应的程序语言。

[0086] 本实施例通过将自然语言语句中的实体类型提取出来,并根据提取的实体类型生成自然语言模板,然后将自然语言模板输入到训练好的神经网络中,将自然语言模板通过拟合计算以及迭代计算,转换成程序模板,并通过训练设计使填充程序模板中预留了待填充槽,用于将自然语言语句中的实体填充至程序模板中,形成自然语言语句对应的程序语言。上述实体类型为用于表示自然语言语句的主干词语对应的词语类型,比如包括文件名,路径名,绝对路径名、时间等,上述的实体包括文件名,路径名,绝对路径名、时间等分别对应的赋值。上述提取实体类型的过程,为识别语义确定主干词语的过程,即提取出的词保留了原自然语言语句的原始语义表达。上述自然语言模板为上述主干词语分别对应的实体类型,按照在原自然语言语句中的排列次序而形成的序列。然后通过将上述序列输入到训练好的神经网络中,通过编码和解码中的拟合和迭代计算,使自然语言模板逐渐转变为程序模板的序列,并预留了待填充槽。本实施例通过将自然语言语句直接转换为程序语言,实现能够让编程人员使用自然语言来表达预期的操作,方便程序员审查和使用。

[0087] 进一步地,所述填充模块5,包括:

[0088] 确定单元,用于根据预设匹配计算公式确定与第一槽匹配的第一实体,其中,所述第一实体包含于所有所述实体中,所述第一槽包含于所有所述待填充的槽。

[0089] 填充单元,用于若是,则将所述第一实体填充至所述第一槽中。

[0090] 生成单元,用于按照所述第一槽的填充过程完成所述程序模板中的所有所述待填充槽的填充,生成所述程序语言。

[0091] 本实施例通过将实体和槽进行匹配配对,实现将实体准确填充于槽,解决了将正确的输入字符串与输出命令的参数值对齐。本实施例的预设匹配计算公式为通过局部匹配函数改进了k-近邻算法来完成的,由于现有k-近邻算法中的欧式距离等不适用于本申请中的槽和实体的匹配,本申请中的槽和实体无具体对应的向量,本实施例通过将距离最近的实体和槽组成的数据对,拼接为一个长向量,通过计算两个长向量之间的相似度,进而得

到待匹配的槽和实体之间的匹配度。通过上述方式实现槽与实体的准确对齐,提高从自然语言语句到程序语言的准确对应性。

[0092] 进一步地,所述确定单元,包括:

[0093] 计算子单元,用于通过计算公式 $Y(i, j) = \sum_{(c, d) \in NN(i, j, k)} d_{(i, j), (c, d)} \cdot V(c, d)$,分别计算

所有所述实体与所述第一槽之间的匹配分数值,其中 $d_{(i, j), (c, d)} = \cos((h_i, h_j), (h_c, h_d))$,

$V(c, d) = \begin{cases} 1, & \text{if } (e_c, s_d) \\ 0, & \text{otherwise} \end{cases}$, h_i 表示所有所述实体分别对应的向量, h_j 表示所述第一槽对应的

的向量, (e_c, s_d) 为已标注匹配状态的第二实体和第二槽, h_c 表示所述第二实体 e_c 对应的向量, h_d 表示所述第二槽 s_d 对应的向量。

[0094] 第一确定子单元,用于确定所述匹配分数值最大时,与所述第一槽对应的指定实体。

[0095] 第二确定子单元,用于将所述指定实体确定为所述第一实体。

[0096] 本实施例中,通过局部匹配函数改进了k-近邻算法,根据k-近邻算法通过计算在相邻区域内出现概率,来判断相似性的原理,改进成通过选择距离待匹配实体和槽数据对最近的已标记匹配数据的实体和槽数据对,并计算上述两对实体和槽数据对的匹配分数值,来克服现有实体和槽之间无法直接实现准确映射向量的问题,且通过改进的k-近邻算法,减少了计算量且提高了实体和槽的匹配准确度。举例地,上述第二实体 e_c 和第二槽 s_d ,分别在Encoder模型和Decoder模型里对应一个隐藏层变量 h_c, h_d ,通过将变量拼接,形成长向量 (h_c, h_d) ,组成了参照实体和槽数据对,且标识了匹配状态,匹配标识为1,比匹配标识为0。通过已确定的待填充槽,即第一槽,分别匹配所有的实体,组成多个包括第一槽的实体和槽数据对,同样 h_i 表示所有实体分别在Encoder模型里对应的隐藏层变量, h_j 表示所述第一槽Decoder模型里对应的隐藏层变量,通过拼接形成长向量 (h_i, h_j) ,通过筛选距离待分析的第一槽的实体和槽数据对,距离值最近的已标注匹配状态的第二实体和第二槽,并计算与已标注匹配状态的第二实体和第二槽的匹配分数值,匹配分数值越大,则表明待分析的第一槽的实体和槽数据对,与已标注匹配状态的第二实体和第二槽越相似。比如已标注匹配状态的第二实体和第二槽标注为匹配,则跟它相似的待分析的第一槽的实体和槽数据对也推定为匹配,实现实体和槽的准确对应。

[0097] 进一步地,所述神经网络模型包括编码-解码模型,转换模块4,包括:

[0098] 拟合单元,用于将所述自然语言模板输入预训练的解码编码模型的隐藏层中,进行拟合得到拟合态自然语言模板并输出,其中,所述解码编码模型的隐藏层包含第一指定数量的形态,所述第一指定数量与所述自然语言模板中包含的实体类型的数量相同。

[0099] 输入单元,用于将所述拟合态自然语言模板输入预训练的解码模型的隐藏层,通过迭代输入计算转换为已排序的所述程序语言模板的列表并输出,其中所述程序语言模板按照预设格式保留所述待填充槽。

[0100] 本实施例通过Encoder的隐藏层实现对自然语言模板的降维拟合,降低处理过程中的复杂度,且提高了对自然语言模板的提取特征的正确识别率。本实施例中Encoder包括4层隐藏层对应7个拟合形态依次为H1至H7,将每层LSTM的步长设置为20;在MSCOCO数据集上,其隐层输出设置为1000,在Flickr30K数据集上,LSTM隐层输出为512,在Flickr8K上,隐

层输出设置为256,以提高降维效果。本实施例的Decoder的隐藏层通过接收Encoder输出的最后一个隐藏层拟合形态作为输入,经过多次迭代输入输出,得到已排序的所述程序语言模板的列表,并按照预设模式保留了待填充槽。上述的Encoder的隐藏层和Decoder的隐藏层相连,且均通过已标记的训练样本训练出最佳参量,使Encoder-Decoder模型输出的程序模板无限接近预设模式。

[0101] 进一步地,所述输入单元,包括:

[0102] 第一输入子单元,用于将所述拟合态自然语言模板作为第一输入数据,输入到解码模型的隐藏层的第一梯度层,确定权重系数最高的向量对应的第一实体类型并输出。

[0103] 第二输入子单元,用于将所述第一实体类型与所述第一梯度层的计算数据作为第二输入数据,输入到解码模型的隐藏层的第二梯度层,确定权重系数次高的向量对应的第二实体类型并输出,其中所述第一梯度层和第二梯度层为所述解码模型的隐藏层中相邻连接的迭代层。

[0104] 第三输入子单元,用于按照由所述第一输入数据到所述第二输入数据的迭代过程,依次迭代输入至所述解码模型的隐藏层,至确定出所述自然语言语句的所有主干词语,其中各所述主干词语分别根据各自对应的权重系数的排序确定,并根据权重系数的排序,形成排序的所述程序语言模板的列表。

[0105] 本实施例通过多个相互连接的梯度层,进行迭代输入输出,比如包括H1`至H5`,将Encoder输出的最后一个隐藏层拟合形态,即拟合态自然语言模板作为第一输入数据,输入到Decoder的隐藏层的H1`,输出对自然语言语句的原语义最重要的第一实体类型Y1`,然后将Y1`以及计算提取Y1`的过程数据同时作为H2`的输入,输出对自然语言语句的原语义次重要的第二实体类型Y2`,然后将Y2`以及计算提取Y2`的过程数据同时作为H3`的输入,依次迭代进行计算提取,直至输出包括预设数量的所有实体类型的所述程序语言模板的列表。举例地,用户输入“查找15天前的日志文件”,对应的自然语言模板为:find all[file name]files older than[Timespan],经过Encoder拟合输出后,经过Decoder的隐藏层的迭代输入输出计算,得到的排序的所述程序语言模板的列表为:Y1:find[path]-name [regex]-mtime[+Timespan];Y2:find[path]-type f-name[regex]-mtime[+Timespan];Y3:find[path]-type f-perm[permission];Y4:find[path]-name[regex]-mtime[+Timespan]xargs ls;Y5:....

[0106] 进一步地,所述区分模块2,包括:

[0107] 调用单元,用于调用根据所述程序模板的构成要素预配置的正则表达式,其中所述正则表达式的类型至少包括两种。

[0108] 判断单元,用于根据第一正则表达式判断所述自然语言语句中,是否包含与所述第一正则表达式适配的指定实体类型,其中,所述第一正则表达式包含于所有所述正则表达式中。

[0109] 提取单元,用于若是,则通过所述第一正则表达式提取所述指定实体类型对应的实体。

[0110] 本实施例通过匹配不同的正则表达式来识别不同的实体类型,并提取相应的实体,方法简便,计算量小且识别提取实体的准确度高。通过将自然语言语句依次输入到各正则表达式中,以判断自然语言语句中是否包括与当前正则表达式匹配的实体类型,若存在,

则提取出该实体。本实施例的多个正则表达式的排布可预先根据各实体类型的重要程序，依次降序排布，以便识别的实体类型按照重要次序形成排列。

[0111] 进一步地，所述匹配模块3，包括：

[0112] 获取单元，用于分别获取所述模式类实体对应的实体和实体数量，以及所述数量类实体对应的实体和实体数量。

[0113] 形成单元，用于根据所述模式类实体对应的实体和实体数量，以及所述数量类实体对应的实体和实体数量，按照预设次序排列所述模式类实体对应的实体和以及所述数量类实体对应的实体，形成所述自然语言模板。

[0114] 本实施例的实体类型包括多个模式类实体和多个数量类实体，比如包括5个模式类实体，包括：“File:file name;Directory:directory name;Path:absolute path;Date/Time:date and time expression;Regex:other pattern arguments”，以及3个数量类实体，包括“Number:number;Size:file size;Timespan:time duration”，以尽量全面覆盖自然语言语句的所有主干词语。通过确定不同实体类型，以及各不同实体类型对应自然语言语句的重要性排序，形成自然语言模板，同时确定并提取各不同实体类型分别对应的实体，作为待填充槽的赋值。比如，用户输入“查找15天前的日志文件，即find all log files older than 15days”，主干词语包括“file name:log;Timespan:15days”，对应的自然语言模板为：find all[file name]files older than[Timespan]。

[0115] 参照图3，本申请实施例中还提供一种计算机设备，该计算机设备可以是服务器，其内部结构可以如图3所示。该计算机设备包括通过系统总线连接的处理器、存储器、网络接口和数据库。其中，该计算机设计的处理器用于提供计算和控制能力。该计算机设备的存储器包括非易失性存储介质、内存储器。该非易失性存储介质存储有操作系统、计算机程序和数据库。该内存储器为非易失性存储介质中的操作系统和计算机程序的运行提供环境。该计算机设备的数据库用于存储地址匹配的过程需要的所有数据。该计算机设备的网络接口用于与外部的端通过网络连接通信。该计算机程序被处理器执行时以实现自然语言转换成程序语言的方法。

[0116] 上述处理器执行上述自然语言转换成程序语言的方法，包括：获取用户输入的自然语言语句；区分所述自然语言语句中包括的所有实体类型，并提取所有所述实体类型分别对应的实体；根据所有所述实体类型匹配自然语言模板；将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板，其中，所述程序语言模板携带待填充槽；将所有所述实体分别对应填充至所述程序模板中的所述待填充槽，生成所述自然语言语句对应的程序语言。

[0117] 上述计算机设备，通过将自然语言语句中的实体类型提取出来，并根据提取的实体类型生成自然语言模板，然后将自然语言模板输入到训练好的神经网络中，将自然语言模板通过拟合计算以及迭代计算，转换成程序模板，并通过训练设计使填充程序模板中预留了待填充槽，用于将自然语言语句中的实体填充至程序模板中，形成自然语言语句对应的程序语言。通过将自然语言语句直接转换为程序语言，实现能够让编程人员使用自然语言来表达预期的操作，方便程序员审查和使用。通过局部匹配函数改进了k-近邻算法来完成，解决了现有k-近邻算法中的欧式距离等不适用于本申请中的槽和实体的匹配的问题，通过将距离最近的实体和槽组成的数据对，拼接为一个长向量，通过计算两个长向量之间

的相似度,进而得到待匹配的槽和实体之间的匹配度。通过上述方式实现槽与实体的准确对齐,提高从自然语言语句到程序语言的准确对应性。通过匹配不同的正则表达式来识别不同的实体类型,并提取相应的实体,方法简便,计算量小且识别提取实体的准确度高。通过将自然语言语句依次输入到各正则表达式中,以判断自然语言语句中是否包括与当前正则表达式匹配的实体类型,若存在,则提取出该实体。

[0118] 在一个实施例中,上述处理器将所有所述实体分别对应填充至所述程序模板中的所述待填充槽,生成所述自然语言语句对应的程序语言的步骤,包括:根据预设匹配计算公式确定与第一槽匹配的第一实体,其中,所述第一实体包含于所有所述实体中,所述第一槽包含于所有所述待填充的槽;若是,则将所述第一实体填充至所述第一槽中;按照所述第一槽的填充过程完成所述程序模板中的所有所述待填充槽的填充,生成所述程序语言。

[0119] 在一个实施例中,上述处理器根据预设匹配计算公式确定与第一槽匹配的第一实体的步骤,包括:通过计算公式
$$Y(i, j) = \sum_{(c, d) \in NN(i, j, k)} d_{(i, j), (c, d)} \cdot V(c, d)$$
,分别计算所有所述

实体与所述第一槽之间的匹配分数值,其中 $d_{(i, j), (c, d)} = \cos((h_i, h_j), (h_c, h_d))$,

$$V(c, d) = \begin{cases} 1, & \text{if } (e_c, s_d) \\ 0, & \text{otherwise} \end{cases}$$
, h_i 表示所有所述实体分别对应的向量, h_j 表示所述第一槽对应的

的向量, (e_c, s_d) 为已标注匹配状态的第二实体和第二槽, h_c 表示所述第二实体 e_c 对应的向量, h_d 表示所述第二槽 s_d 对应的向量;确定所述匹配分数值最大时,与所述第一槽对应的指定实体;将所述指定实体确定为所述第一实体。

[0120] 在一个实施例中,所述神经网络模型包括编码-解码模型,上述处理器将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板的步骤,包括:将所述自然语言模板输入预训练的解码编码模型的隐藏层中,进行拟合得到拟合态自然语言模板并输出,其中,所述解码编码模型的隐藏层包含第一指定数量的形态,所述第一指定数量与所述自然语言模板中包含的实体类型的数量相同;将所述拟合态自然语言模板输入预训练的解码模型的隐藏层,通过迭代输入计算转换为已排序的所述程序语言模板的列表并输出,其中所述程序语言模板按照预设格式保留所述待填充槽。

[0121] 在一个实施例中,上述处理器将所述拟合态自然语言模板输入预训练的解码模型的隐藏层,通过迭代输入计算转换为已排序的所述程序语言模板的列表并输出的步骤,包括:将所述拟合态自然语言模板作为第一输入数据,输入到解码模型的隐藏层的第一梯度层,确定权重系数最高的向量对应的第一实体类型并输出;将所述第一实体类型与所述第一梯度层的计算数据作为第二输入数据,输入到解码模型的隐藏层的第二梯度层,确定权重系数次高的向量对应的第二实体类型并输出,其中所述第一梯度层和第二梯度层为所述解码模型的隐藏层中相邻连接的迭代层;按照由所述第一输入数据到所述第二输入数据的迭代过程,依次迭代输入至所述解码模型的隐藏层,至确定出所述自然语言语句的所有主干词语,其中各所述主干词语分别根据各自对应的权重系数的排序确定,并根据权重系数的排序,形成排序的所述程序语言模板的列表。

[0122] 在一个实施例中,上述处理器区分所述自然语言语句中包括的所有实体类型,并提取所有所述实体类型分别对应的实体的步骤,包括:调用根据所述程序模板的构成要素预配置的正则表达式,其中所述正则表达式的类型至少包括两种;根据第一正则表达式判

断所述自然语言语句中,是否包含与所述第一正则表达式适配的指定实体类型,其中,所述第一正则表达式包含于所有所述正则表达式中;若是,则通过所述第一正则表达式提取所述指定实体类型对应的实体。

[0123] 在一个实施例中,所述实体类型包括模式类实体和数量类实体,根据所有所述实体类型匹配自然语言模板的步骤,包括:分别获取所述模式类实体对应的实体和实体数量,以及所述数量类实体对应的实体和实体数量;根据所述模式类实体对应的实体和实体数量,以及所述数量类实体对应的实体和实体数量,按照预设次序排列所述模式类实体对应的实体和以及所述数量类实体对应的实体,形成所述自然语言模板。

[0124] 本领域技术人员可以理解,图3中示出的结构,仅仅是与本申请方案相关的部分结构的框图,并不构成对本申请方案所应用于其上的计算机设备的限定。

[0125] 本申请一实施例还提供一种计算机可读存储介质,其上存储有计算机程序,计算机程序被处理器执行时实现自然语言转换成程序语言的方法,包括:获取用户输入的自然语言语句;区分所述自然语言语句中包括的所有实体类型,并提取所有所述实体类型分别对应的实体;根据所有所述实体类型匹配自然语言模板;将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板,其中,所述程序语言模板携带待填充槽;将所有所述实体分别对应填充至所述程序模板中的所述待填充槽,生成所述自然语言语句对应的程序语言。

[0126] 上述计算机可读存储介质,通过将自然语言语句中的实体类型提取出来,并根据提取的实体类型生成自然语言模板,然后将自然语言模板输入到训练好的神经网络中,将自然语言模板通过拟合计算以及迭代计算,转换成程序模板,并通过训练设计使填充程序模板中预留了待填充槽,用于将自然语言语句中的实体填充至程序模板中,形成自然语言语句对应的程序语言。通过将自然语言语句直接转换为程序语言,实现能够让编程人员使用自然语言来表达预期的操作,方便程序员审查和使用。通过局部匹配函数改进了k-近邻算法来完成,解决了现有k-近邻算法中的欧式距离等不适用于本申请中的槽和实体的匹配的问题,通过将距离最近的实体和槽组成的数据对,拼接为一个长向量,通过计算两个长向量之间的相似度,进而得到待匹配的槽和实体之间的匹配度。通过上述方式实现槽与实体的准确对齐,提高从自然语言语句到程序语言的准确对应性。通过匹配不同的正则表达式来识别不同的实体类型,并提取相应的实体,方法简便,计算量小且识别提取实体的准确度高。通过将自然语言语句依次输入到各正则表达式中,以判断自然语言语句中是否包括与当前正则表达式匹配的实体类型,若存在,则提取出该实体。

[0127] 在一个实施例中,上述处理器将所有所述实体分别对应填充至所述程序模板中的所述待填充槽,生成所述自然语言语句对应的程序语言的步骤,包括:根据预设匹配计算公式确定与第一槽匹配的第一实体,其中,所述第一实体包含于所有所述实体中,所述第一槽包含于所有所述待填充的槽;若是,则将所述第一实体填充至所述第一槽中;按照所述第一槽的填充过程完成所述程序模板中的所有所述待填充槽的填充,生成所述程序语言。

[0128] 在一个实施例中,上述处理器根据预设匹配计算公式确定与第一槽匹配的第一实体的步骤,包括:通过计算公式
$$Y(i, j) = \sum_{(c, d) \in NN(i, j, k)} d_{(i, j), (c, d)} \cdot V(c, d)$$
,分别计算所有所述

实体与所述第一槽之间的匹配分数值,其中 $d_{(i, j), (c, d)} = \cos((h_i, h_j), (h_c, h_d))$,

$$V(c, d) = \begin{cases} 1, & \text{if } (e_c, s_d) \\ 0, & \text{otherwise} \end{cases}$$
 h_i 表示所有所述实体分别对应的向量, h_j 表示所述第一槽对应的

的向量, (e_c, s_d) 为已标注匹配状态的第二实体和第二槽, h_c 表示所述第二实体 e_c 对应的向量, h_d 表示所述第二槽 s_d 对应的向量;确定所述匹配分数值最大时,与所述第一槽对应的指定实体;将所述指定实体确定为所述第一实体。

[0129] 在一个实施例中,所述神经网络模型包括编码-解码模型,上述处理器将所述自然语言模板通过预训练的神经网络模型转换成程序语言模板的步骤,包括:将所述自然语言模板输入预训练的解码编码模型的隐藏层中,进行拟合得到拟合态自然语言模板并输出,其中,所述解码编码模型的隐藏层包含第一指定数量的形态,所述第一指定数量与所述自然语言模板中包含的实体类型的数量相同;将所述拟合态自然语言模板输入预训练的解码模型的隐藏层,通过迭代输入计算转换为已排序的所述程序语言模板的列表并输出,其中所述程序语言模板按照预设格式保留所述待填充槽。

[0130] 在一个实施例中,上述处理器将所述拟合态自然语言模板输入预训练的解码模型的隐藏层,通过迭代输入计算转换为已排序的所述程序语言模板的列表并输出的步骤,包括:将所述拟合态自然语言模板作为第一输入数据,输入到解码模型的隐藏层的第一梯度层,确定权重系数最高的向量对应的第一实体类型并输出;将所述第一实体类型与所述第一梯度层的计算数据作为第二输入数据,输入到解码模型的隐藏层的第二梯度层,确定权重系数次高的向量对应的第二实体类型并输出,其中所述第一梯度层和第二梯度层为所述解码模型的隐藏层中相邻连接的迭代层;按照由所述第一输入数据到所述第二输入数据的迭代过程,依次迭代输入至所述解码模型的隐藏层,至确定出所述自然语言语句的所有主干词语,其中各所述主干词语分别根据各自对应的权重系数的排序确定,并根据权重系数的排序,形成排序的所述程序语言模板的列表。

[0131] 在一个实施例中,上述处理器区分所述自然语言语句中包括的所有实体类型,并提取所有所述实体类型分别对应的实体的步骤,包括:调用根据所述程序模板的构成要素预配置的正则表达式,其中所述正则表达式的类型至少包括两种;根据第一正则表达式判断所述自然语言语句中,是否包含与所述第一正则表达式适配的指定实体类型,其中,所述第一正则表达式包含于所有所述正则表达式中;若是,则通过所述第一正则表达式提取所述指定实体类型对应的实体。

[0132] 在一个实施例中,所述实体类型包括模式类实体和数量类实体,根据所有所述实体类型匹配自然语言模板的步骤,包括:分别获取所述模式类实体对应的实体和实体数量,以及所述数量类实体对应的实体和实体数量;根据所述模式类实体对应的实体和实体数量,以及所述数量类实体对应的实体和实体数量,按照预设次序排列所述模式类实体对应的实体和以及所述数量类实体对应的实体,形成所述自然语言模板。

[0133] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,上述的计算机程序可存储于一非易失性计算机可读取存储介质中,该计算机程序在执行时,可包括如上述各方法的实施例的流程。其中,本申请所提供的和实施例中所使用的对存储器、存储、数据库或其它介质的任何引用,均可包括非易失性和/或易失性存储器。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)、电可擦除可编程ROM(EEPROM)或闪存。易失性存储器可包括

随机存取存储器 (RAM) 或者外部高速缓冲存储器。作为说明而非局限, RAM以多种形式可得, 诸如静态RAM (SRAM)、动态RAM (DRAM)、同步DRAM (SDRAM)、双速据率SDRAM (SSRSDRAM)、增强型SDRAM (ESDRAM)、同步链路 (Synchlink) DRAM (SLDRAM)、存储器总线 (Rambus) 直接RAM (RDRAM)、直接存储器总线动态RAM (DRDRAM)、以及存储器总线动态RAM (RDRAM) 等。

[0134] 需要说明的是, 在本文中, 术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含, 从而使得包括一系列要素的过程、装置、物品或者方法不仅包括那些要素, 而且还包括没有明确列出的其他要素, 或者是还包括为这种过程、装置、物品或者方法所固有的要素。在没有更多限制的情况下, 由语句“包括一个……”限定的要素, 并不排除在包括该要素的过程、装置、物品或者方法中还存在另外的相同要素。

[0135] 以上所述仅为本申请的优选实施例, 并非因此限制本申请的专利范围, 凡是利用本申请说明书及附图内容所作的等效结构或等效流程变换, 或直接或间接运用在其他相关的技术领域, 均同理包括在本申请的专利保护范围内。

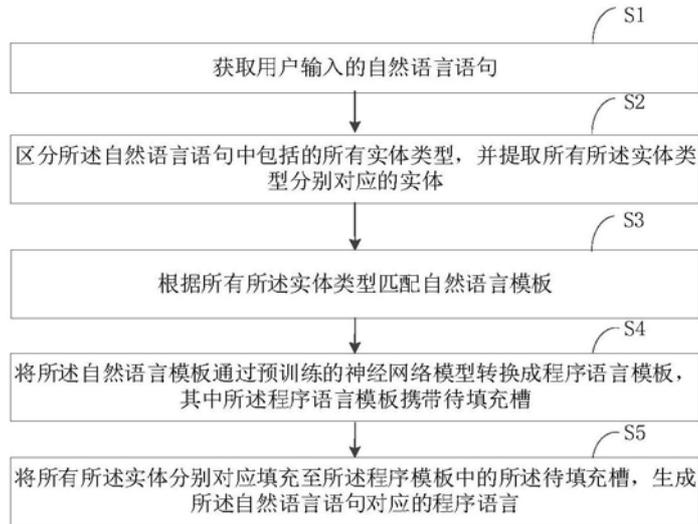


图1

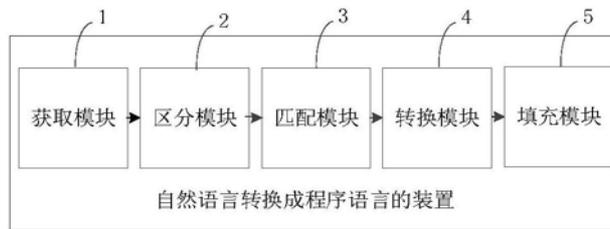


图2

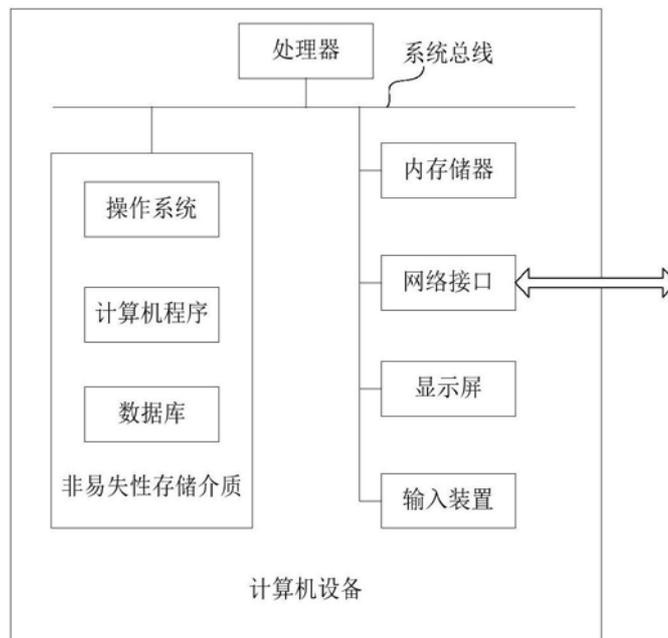


图3