(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 119045799 A
(43)申请公布日 2024.11.29

(21)申请号 202411192586.6

(22)申请日 2024.08.28

(71)申请人 北京航空航天大学
地址 100191 北京市海淀区学院路37号

(72)发明人 吴文峻 杨京波 任健

(74)专利代理机构 北京科迪生专利代理有限责任公司 11251
专利代理师 尹莹莹 顾炜

(51)Int.Cl.
G06F 8/33(2018.01)
G06F 8/71(2018.01)
G06F 8/40(2018.01)
G06N 3/0455(2023.01)
G06N 3/08(2023.01)

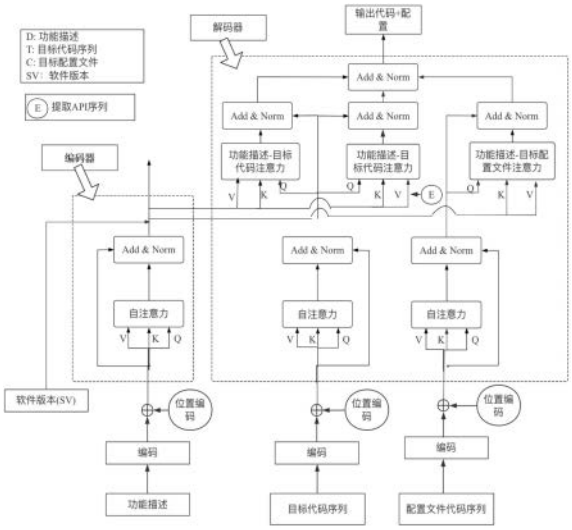
权利要求书3页 说明书5页 附图1页

(54)发明名称

一种基于软件演化的API调用序列代码智能生成方法及装置

(57)摘要

一种基于软件演化的API调用序列代码智能生成方法及装置,该方法采用编码器-解码器的神经网络模型执行,包括:接收想要实现的代码对应的软件版本描述、所需的功能描述;接收原始代码并将其转换为目标代码序列;接收原始代码所需的配置文件并将其转换为配置文件代码序列;分别将功能描述、目标代码序列、以及配置文件代码序列进行编码和位置编码,以及通过注意力模块、add&norm层进行一系列处理,对编码器-解码器的神经网络模型进行训练,直至训练结果收敛,获得基于软件演化的智能生成API调用序列代码的模型,通过该模型获得所需代码。根据本发明技术方案,能够针对指定版本软件,根据用户需求描述,生成满足用户需求的高质量API调用序列代码。



CN 119045799 A

1.一种基于软件演化的API调用序列代码智能生成方法,该方法采用编码器-解码器的神经网络模型执行,其特征在于,包括以下步骤:

步骤1,接收用户输入,用户输入包括所需代码对应的软件版本描述、所需的功能描述;

步骤2,接收原始代码并将其转换为目标代码序列,目标代码序列指的是将原始代码转换为1个行向量表示,行向量表示中的每个元素与原始代码中的每个元素一一对应;

步骤3,接收原始代码所需的配置文件并将其转换为配置文件代码序列,配置文件代码序列为将原始代码所需的配置文件转换为一个行向量表示,行向量表示中的每个元素与原始配置文件中的元素一一对应;

步骤4,分别将功能描述、目标代码序列、以及配置文件代码序列进行编码,维度为自定义;

步骤5:分别将功能描述、目标代码序列、以及配置文件代码序列进行位置编码;

步骤6:通过自注意力模块,分别将步骤4、5中的编码向量合并;

步骤7:根据公式(1)以及残差连接&归一化层对步骤6的结果进行计算,以得到功能描述计算结果,其中,先根据公式(1)进行计算,之后进行add&norm计算:

$$\mathrm{attention}(Q,K,V)=\mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V \qquad (1)$$

在公式(1)中,softmax为归一化指数函数,d_k是矩阵K的列数,Q,K,V分别是注意力模块的Q输入、K输入和V输入;

步骤8:将步骤1接收的软件版本描述与步骤7中的功能描述计算结果合并;

步骤9:将步骤8中的结果看作K与V,步骤7中的功能描述计算结果作为Q,通过公式(1)进行D-T注意力计算,D-T 注意力计算指的是功能描述-目标代码序列注意力计算;

步骤10:将步骤9的计算结果再次进行Add & Norm层计算;

步骤11:将步骤8中的结果一个作为K,提取步骤8结果中的应用编程接口API作为V,步骤7中的功能描述计算结果作为Q,通过公式(1)进行D-T 注意力计算;

步骤12:对步骤11中的计算结果再次进行Add & Norm层计算;

步骤13:将步骤8中的计算结果作为K与V,步骤7中的计算结果里面的配置文件代码结果作为Q,通过公式(1)进行D-C注意力计算,D-C注意力计算指的是功能描述-目标配置文件注意力计算;

步骤14:对步骤13中的结果进行Add & Norm层计算;

步骤15:步骤14、12、10的结果进行Add & Norm层计算;

步骤16:重复步骤1-15,对所述编码器-解码器的神经网络模型进行训练,直至训练结果收敛;

步骤17:步骤16结束后,获得基于软件演化的智能生成API调用序列代码的模型;

步骤18:基于从用户接收的输入软件版本及功能描述,通过步骤17得到的模型,获得所需代码。

2.一种基于软件演化的API调用序列代码智能生成装置,其特征在于,包括:

接收模块,用于接收用户输入,用户输入包括所需代码对应的软件版本描述、所需的功能描述;

第一转换模块,用于接收原始代码并将其转换为目标代码序列,目标代码序列指的是

将原始代码转换为1个行向量表示,行向量表示中的每个元素与原始代码中的每个元素一一对应;

第二转换模块,用于接收原始代码所需的配置文件并将其转换为配置文件代码序列,配置文件代码序列为将原始代码所需的配置文件转换为一个行向量表示,行向量表示中的每个元素与原始配置文件中的元素一一对应;

编码模块,用于分别将功能描述、目标代码序列、以及配置文件代码序列进行编码,维度为自定义;

位置编码模块,用于分别将功能描述、目标代码序列、以及配置文件代码序列进行位置编码;

自注意力模块,用于通过自注意力模块,分别将编码模块和位置编码模块中的编码向量合并;

功能描述计算模块,用于根据公式(1)以及残差连接&归一化层对自注意力模块的结果进行计算,以得到功能描述计算结果:

$$\text{attention}(Q,K,V)=\text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

在公式(1)中,softmax为归一化指数函数,d_k是矩阵K的列数,Q,K,V分别是注意力模块的Q输入、K输入和V输入;

合并模块,用于将接收模块接收的软件版本描述与功能描述计算模块的功能描述计算结果合并;

第一D-T注意力模块,将合并模块的结果看作K和V,功能描述计算模块中功能描述计算结果作为Q,通过公式(1)进行D-T注意力计算,D-T注意力计算指的是功能描述-目标代码序列注意力计算;

第一残差连接&归一化层模块,用于将第一D-T注意力模块的计算结果再次进行残差连接&归一化层计算;

第二D-T注意力模块,用于将合并模块的结果作为K,提取合并模块的结果中的API作为V,功能描述计算模块中功能描述计算后的结果作为Q,通过公式(1)进行D-T注意力计算;

第二残差连接&归一化层模块,用于对第二D-T注意力模块中的计算结果再次进行残差连接&归一化层计算;

D-C注意力模块,用于将合并模块中的计算结果作为K与V,功能描述计算模块中的计算结果里面的配置文件代码结果作为Q,通过公式(1)进行D-C注意力计算;D-C注意力计算指的是功能描述-目标配置文件注意力计算;

第三残差连接&归一化层模块,对D-C注意力模块中的结果进行残差连接&归一化层计算;

第四残差连接&归一化层模块,用于对第三残差连接&归一化层模块、第二残差连接&归一化层模块、第一残差连接&归一化层模块的结果进行残差连接&归一化层计算;

迭代模块,用于重复以上模块的执行过程,对所述模型进行训练,直至训练结果收敛;

模型生成模块,用于根据以上模块的执行结果,获得基于软件演化的智能生成API调用序列代码的模型;

代码生成模块,用于基于从用户接收的输入软件版本及功能描述,通过模型生成模块

获得的模型,获得所需代码。

一种基于软件演化的API调用序列代码智能生成方法及装置

技术领域

[0001] 本发明涉及计算机及人工智能领域、软件工程领域,尤其涉及一种基于软件演化的API调用序列代码智能生成方法及装置。

背景技术

[0002] 人工智能的发展给软件开发及维护带来很大便利,例如自动生成或补全代码,对代码进行纠错等,然而在实际软件开发及维护业务中,大多数人工智能技术生成代码的质量难以保证,同时,由于软件版本不断进行演化,生成的代码需要及时根据软件版本进行更新,例如Android 6与Android 4中,关于位置定位功能的代码存在很大差别,这是由于Android 6引入了动态权限申请机制,导致代码中需要添加申请定位权限的代码才能进行定位功能,并且在Android功能代码中,实现定位功能,需要调用申请权限、弹出权限框、处理返回数据等多个相关的应用编程接口(API),一系列的API构成了API调用序列。在目前的生成代码方法中,很少有专门基于软件版本演化生成API调用序列代码的方法,这在实际代码开发及维护中带来一定弊端。

发明内容

[0003] 为解决或至少缓解以上技术问题,作出本发明。

[0004] 根据本发明第一方面,提供一种基于软件演化的API调用序列代码的智能生成方法,该方法采用编码器-解码器的神经网络模型执行,包括以下步骤:

步骤1,接收用户输入,用户输入包括想要实现的代码对应的软件版本描述、所需的功能描述;

步骤2,接收原始代码并将其转换为目标代码序列,目标代码序列指的是将原始代码转换为1个行向量表示,行向量表示中的每个元素与原始代码中的每个元素一一对应;

步骤3,接收原始代码所需的配置文件并将其转换为配置文件代码序列,配置文件代码序列为将原始代码所需的配置文件转换为一个行向量表示,行向量表示中的每个元素与原始配置文件中的元素一一对应;

步骤4,分别将功能描述、目标代码序列、以及配置文件代码序列进行编码,维度为自定义;

步骤5:分别将功能描述、目标代码序列、以及配置文件代码序列进行位置编码;

步骤6:通过自注意力模块,分别将步骤4、5中的编码向量合并;

步骤7:根据公式(1)以及残差连接&归一化层对步骤6的结果进行计算,以得到功能描述计算结果,其中,先根据公式(1)进行计算,之后进行add&norm计算:

$$\mathrm{attention}(Q,K,V)=\mathrm{softmax}(\frac{QK^{T}}{\sqrt{d_k}})V \qquad (1)$$

在公式(1)中,softmax为归一化指数函数,d_k是矩阵K的列数,Q,K,V分别是注意力模块的Q输入、K输入和V输入;

步骤8:将步骤1接收的软件版本描述与步骤7中的功能描述计算结果合并;

步骤9:将步骤8中的结果看作K与V,步骤7中的功能描述计算结果作为Q,通过公式(1)进行D-T注意力计算,D-T 注意力计算指的是功能描述-目标代码序列注意力计算;

步骤10:将步骤9的计算结果再次进行Add & Norm层计算;

步骤11:将步骤8中的结果一个作为K,提取步骤8结果中的应用编程接口API作为V,步骤7中的功能描述计算结果作为Q,通过公式(1)进行D-T注意力计算;

步骤12:对步骤11中的计算结果再次进行Add & Norm层计算;

步骤13:将步骤8中的计算结果作为K与V,步骤7中的计算结果里面的配置文件代码结果作为Q,通过公式(1)进行D-C注意力计算,D-C注意力计算指的是功能描述-目标配置文件注意力计算;

步骤14:对步骤13中的结果进行Add & Norm层计算;

步骤15:步骤14、12、10的结果进行Add & Norm层计算;

步骤16:重复步骤1-15,对所述编码器-解码器的神经网络模型进行训练,直至训练结果收敛;

步骤17:步骤16结束后,获得基于软件演化的智能生成API调用序列代码的模型;

步骤18:基于从用户接收的输入软件版本及功能描述,通过步骤17得到的模型,获得所需代码。

[0005] 有益效果:

通过本发明的方法及装置,开发人员通过指定软件版本及描述功能需求,可以有效获取所需代码,从而有助于软件开发及维护,该方法可以适用多种编程,例如Java编程或者Android编程等。

附图说明

[0006] 图1示出了根据本发明实施方式的基于软件演化的API调用序列代码智能生成方法的模型图。

[0007] 图2示出了根据本发明实施方式的基于软件演化的API调用序列代码智能生成装置。

具体实施方式

[0008] 下面将参照附图更详细地描述本公开的示例性实施方式。虽然附图中显示了本公开的示例性实施方式,然而应当理解,可以以各种形式实现本公开,且本公开不应被这里阐述的实施方式所限制。相反,提供这些实施方式是为了能够更透彻地理解本公开,并且能够将本公开的范围完整地传达给本领域的技术人员。

[0009] 实施方式一:

根据本发明实施方式,提供一种基于软件演化的API调用序列代码智能生成方法。该方法采用编码器-解码器的神经网络模型,模型如图1所示。

[0010] 其中,D表示功能描述,用以指明用户需要实现的目标功能,SV表示软件版本,T表示目标代码序列,C表示目标配置文件,E表示提取API序列,$Q \in R^{N \times T_q \times d_{model}}$,$K \in R^{N \times T_k \times d_{model}}$,$V \in R^{N \times T_k \times d_{model}}$,R表示多维度的张量,N为模型训练时的批量数据大小

值，d_{model}是模型训练时将数据编码的维度大小值，T_q和T_k分别为张量第二个维度的大小值，K、V、Q是注意力模型的常见输入，图中自注意力(Self-Attention)模块的计算公式如下：

$$attention(Q,K,V)=softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

在公式(1)中，softmax为归一化指数函数，d_k是矩阵K的列数。D-T 注意力模块与D-C注意力模块的计算原理一致。其中，D-T注意力模块指的是功能描述-目标代码序列注意力模块。D-C注意力模块指的是功能描述-目标配置文件注意力模块。

[0011] 图1中，Add & Norm 表示残差连接和层归一化，Add具体地可以表示为 X+F(X)。X为一变量，F(X)表示经函数变换后的取值，Norm表示神经网络训练时的层归一化(Layer Normalization)。

[0012] 下面参考图1示出的基于软件演化的API调用序列代码的智能生成模型来描述该智能生成方法。该智能生成方法包括以下步骤：

步骤1，接收用户输入，用户输入包括所需代码对应的软件版本描述、所需的功能描述。软件版本描述可以包含文字描述形式：例如，软件版本号为 1.0，如果是Android等软件，可以描述为：该软件版本基于Android 14等等，功能描述包括该软件需要实现的步骤及具体功能，例如：该软件首先在启动页面设置一个带有列表框的布局文件，点击列表框里面的一项，可以进入到查看该列表项具体详情的页面。

[0013] 步骤2，接收原始代码并将其转换为目标代码序列，目标代码序列指的是将原始代码转换为1个行向量表示，行向量表示中的每个元素与原始代码中的每个元素一一对应。

[0014] 步骤3，接收原始代码所需的配置文件并将其转换为配置文件代码序列，配置文件代码序列为将原始代码所需的配置文件转换为一个行向量表示，行向量表示中的每个元素与原始配置文件中的元素一一对应。

[0015] 步骤4，分别将功能描述、目标代码序列、以及配置文件代码序列进行编码，维度为自定义。编码指的是将文字输入转换为不同维度的数字表示的张量，例如使用python编程编程语言中的torch.nn嵌入工具，将文字转换为张量。

[0016] 步骤5：分别将功能描述、目标代码序列、以及配置文件代码序列进行位置编码。

[0017] 位置编码的过程包括执行以下两个公式，分别为：

$$PositionEmbedding_{pos,2i}=\sin(\frac{pos}{10000^{\frac{2i}{d_{model}}}}) \quad (2)$$

[0018] $$PositionEmbedding_{pos,2i+1}=\cos(\frac{pos}{10000^{\frac{2i}{d_{model}}}}) \quad (3)$$

其中，pos表示一句话中一个标记(token)的位置，每个token的位置编码是一个向量，i表示这个向量中每个元素的位置，d_{model}表示位置编码向量的维度，positionembedding表示位置编码；

步骤6：通过自注意力模块，分别将步骤4、5中的编码向量合并。

[0019] 步骤7：根据公式(1)以及残差连接&归一化(Add & Norm)层对步骤6的结果进行计算，以得到功能描述计算结果。其中，先根据公式(1)进行计算，之后进行add&norm计算。

[0020] 步骤8：将步骤1接收的软件版本描述与步骤7中的功能描述计算结果合并。

[0021] 步骤9:步骤8中的结果作为看作K与V,步骤7中功能描述计算结果作为Q,通过公式(1)进行D-T注意力计算。

[0022] 步骤10:将步骤9的计算结果再次进行Add & Norm层计算。

[0023] 步骤11:将步骤8中的结果一个作为K,提取步骤8结果中的API作为V,步骤7中功能描述计算后的结果作为Q,通过公式(1)进行D-T 注意力计算。

[0024] 步骤12:对步骤11中的计算结果再次进行Add & Norm层计算。

[0025] 步骤13:将步骤8中的计算结果作为K与V,步骤7中的计算结果里面的配置文件代码结果作为Q,通过公式(1)进行D-C Attention计算。

[0026] 步骤14:对步骤13中的结果进行Add & Norm层计算。

[0027] 步骤15:步骤14、12、10的结果进行Add & Norm层计算。

[0028] 步骤16:重复步骤1-15,对所述编码器-解码器的神经网络模型进行训练,直至训练结果收敛。判断训练结果收敛的方式是模型最后的一系列输出结果中,每相邻两个数据之差在一个预设范围内,例如小于0.0002。

[0029] 步骤17:步骤16结束后,获得基于软件演化的智能生成API调用序列代码的模型。

[0030] 步骤18:基于从用户接收的输入软件版本及功能描述,通过步骤17得到的模型,获得所需代码。

[0031] 实施方式二:

根据本发明实施方式,提供一种基于软件演化的API调用序列代码智能生成装置。图2展示了本发明的一种基于软件演化的智能生成API调用序列代码的装置。在图2所示的装置中,用户在输入端输入需求描述及对应软件版本信息,在装置输出端,可以获得所需的API调用序列代码。

[0032] 该装置包括:

接收模块,用于接收用户输入,用户输入包括所需代码对应的软件版本描述、所需的功能描述。

[0033] 第一转换模块,用于接收原始代码并将其转换为目标代码序列,目标代码序列指的是将原始代码转换为1个行向量表示,行向量表示中的每个元素与原始代码中的每个元素一一对应。

[0034] 第二转换模块,用于接收原始代码所需的配置文件并将其转换为配置文件代码序列,配置文件代码序列为将原始代码所需的配置文件转换为一个行向量表示,行向量表示中的每个元素与原始配置文件中的元素一一对应。

[0035] 编码模块,用于分别将功能描述、目标代码序列、以及配置文件代码序列进行编码,维度为自定义。

[0036] 位置编码模块,用于分别将功能描述、目标代码序列、以及配置文件代码序列进行位置编码。

[0037] 自注意力模块,用于通过自注意力模块,分别将编码模块和位置编码模块中的编码向量合并。

[0038] 功能描述计算模块,用于根据公式(1)以及残差连接&归一化(Add & Norm)层对自注意力模块的结果进行计算,以得到功能描述计算结果,该功能描述计算结果、配置文件代码结果。

[0039] 合并模块,用于将接收模块接收的软件版本描述与功能描述计算模块的功能描述计算结果合并。

[0040] 第一D-T 注意力模块,将合并模块的结果看作K和V,功能描述计算模块中功能描述计算结果作为Q,通过公式(1)进行D-T注意力计算,D-T 注意力模块指的是功能描述-目标代码序列注意力模块。

[0041] 第一残差连接&归一化层模块,用于将第一D-T 注意力模块的计算结果再次进行残差连接&归一化层计算。

[0042] 第二D-T注意力模块,用于将合并模块的结果作为K,提取合并模块的结果中的API作为V,功能描述计算模块中功能描述计算后的结果作为Q,通过公式(1)进行D-T注意力计算。

[0043] 第二残差连接&归一化层模块,用于对第二D-T注意力模块中的计算结果再次进行残差连接&归一化层计算。

[0044] D-C注意力模块,用于将合并模块中的计算结果作为K与V,功能描述计算模块中的计算结果里面的配置文件代码结果作为Q,通过公式(1)进行D-C 注意力计算;D-C注意力模块指的是功能描述-目标配置文件注意力模块。

[0045] 第三残差连接&归一化层模块,对D-C注意力模块中的结果进行残差连接&归一化层计算。

[0046] 第四残差连接&归一化层模块,用于对第三残差连接&归一化层模块、第二残差连接&归一化层模块、第一残差连接&归一化层模块的结果进行残差连接&归一化层计算。

[0047] 迭代模块,用于重复以上模块的执行过程,对所述模型进行训练,直至训练结果收敛。

[0048] 模型生成模块,用于根据以上模块的执行结果,获得基于软件演化的智能生成API调用序列代码的模型。

[0049] 代码生成模块,用于基于从用户接收的输入软件版本及功能描述,通过模型生成模块获得的模型,获得所需代码。

[0050] 以上装置未详述部分,还请参考关于方法实施方式的描述。

[0051] 在此处所提供的说明书中,说明了大量具体细节。然而,能够理解,本发明的实施方式可以在没有这些具体细节的情况下被实践。在一些实例中,并未详细示出公知的方法、结构和技术,以便不模糊对本说明书的理解。

[0052] 尽管根据有限数量的实施方式描述了本发明,但是受益于上面的描述,本技术领域内的技术人员明白,在由此描述的本发明的范围内,可以设想其它实施方式。此外,应当注意,本说明书中使用的语言主要是为了可读性和教导的目的而选择的,而不是为了解释或者限定本发明的主题而选择的。

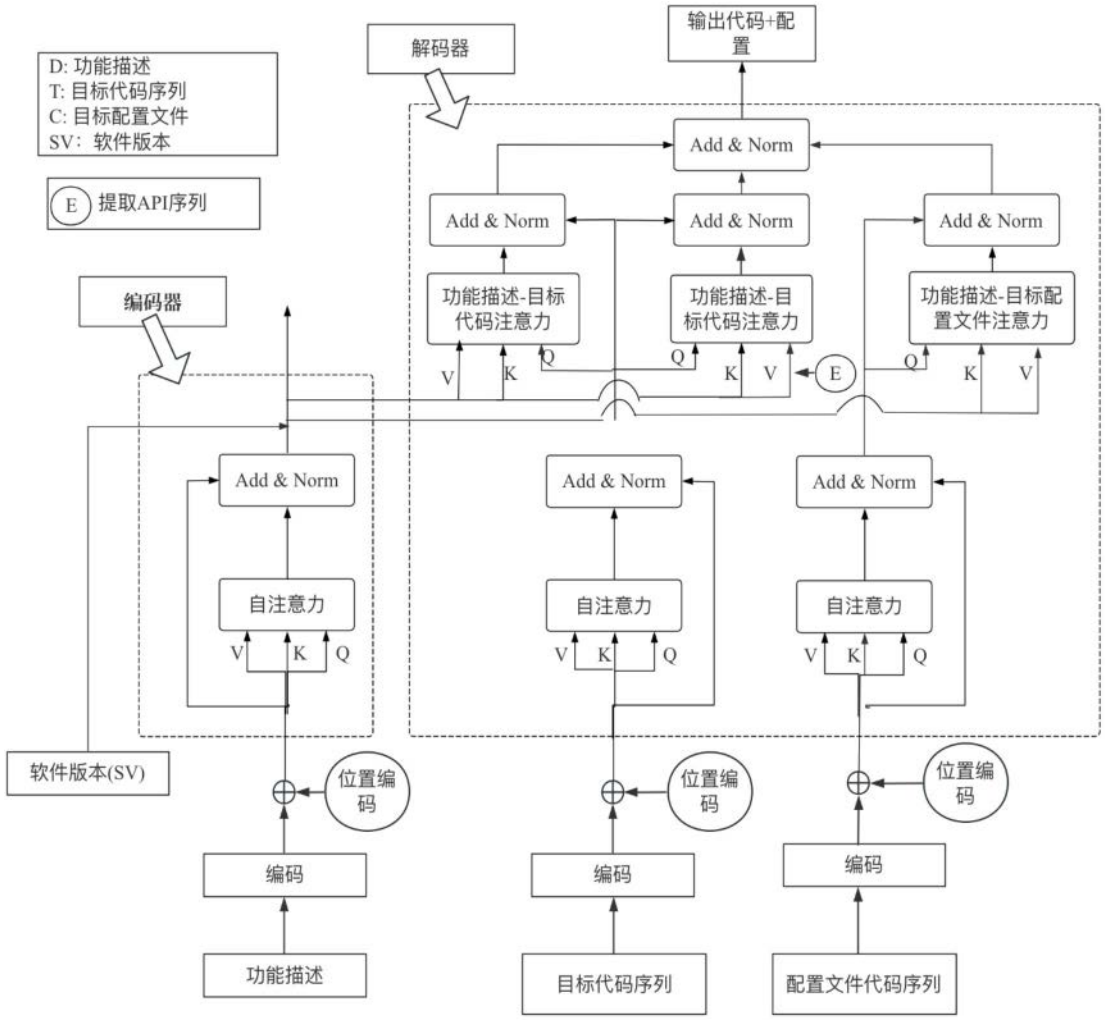

D: 功能描述
T: 目标代码序列
C: 目标配置文件
SV: 软件版本
E 提取API序列
解码器
输出代码+配置
Add & Norm
功能描述-目标代码注意力
功能描述-目标代码注意力
功能描述-目标配置文件注意力
编码器
自注意力
软件版本(SV)
位置编码
编码
功能描述
目标代码序列
配置文件代码序列

图1

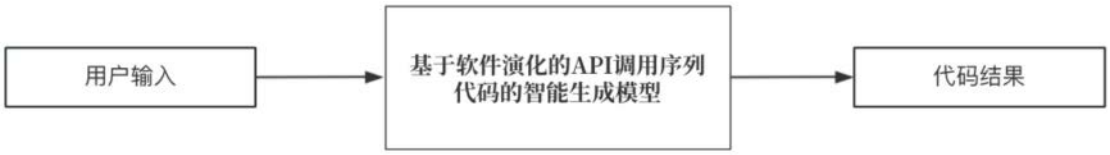

用户输入
基于软件演化的API调用序列代码的智能生成模型
代码结果

图2