



(12) 发明专利申请

(10) 申请公布号 CN 117251553 A

(43) 申请公布日 2023. 12. 19

(21) 申请号 202311516989.7

G06N 3/08 (2023.01)

(22) 申请日 2023.11.15

(71) 申请人 知学云(北京)科技股份有限公司
地址 100071 北京市丰台区西四环南路101号6层637号

(72) 发明人 赵隽隽 潘斌 赵剑飞 欧阳禄萍 张怀仁

(74) 专利代理机构 北京力量专利代理事务所
(特殊普通合伙) 11504
专利代理师 李萱

(51) Int. Cl.
G06F 16/332 (2019.01)
G06F 40/30 (2020.01)
G06F 40/289 (2020.01)
G06N 3/0464 (2023.01)

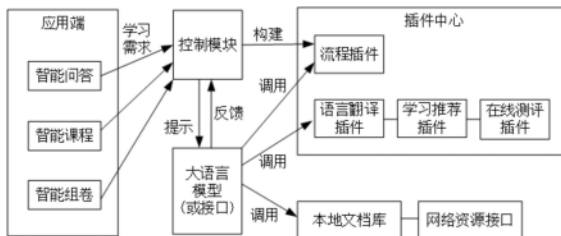
权利要求书3页 说明书11页 附图3页

(54) 发明名称

基于自定义插件和大语言模型的智能学习交互方法

(57) 摘要

本发明实施例公开了一种基于自定义插件和大语言模型的学习交互方法,包括:响应于特定应用的触发操作,将学习需求发送至大语言模型,并提示大语言模型分析是否存在能够满足学习需求的自定义插件;各自定义插件包括语言翻译插件、学习推荐插件、在线测评插件和流程插件中的至少之一,所述流程插件用于协调其它插件的执行顺序;通过大语言模型对所述学习需求进行语义分析,并与各自定义插件的描述信息进行语义匹配;在匹配成功的情况下,提示大语言模型根据流程插件准备匹配插件的入参并调用API,由所述匹配插件完成学习任务;在未匹配成功的情况下,提示大语言模型根据自建文档库或网络资源完成学习任务。



1. 一种基于自定义插件和大语言模型的学习交互方法,其特征在于,包括:

响应于特定应用的触发操作,将学习需求发送至大语言模型,并提示大语言模型分析是否存在能够满足学习需求的自定义插件;其中,所述特定应用包括智能问答、智能课程和智能组卷中的至少之一;各自定义插件包括语言翻译插件、学习推荐插件、在线测评插件和流程插件中的至少之一,所述流程插件用于协调其它插件的执行顺序;

通过大语言模型对所述学习需求进行语义分析,并与各自定义插件的描述信息进行语义匹配,其中,自定义插件的描述信息包括:插件的入参、出参和API,以及插件的作用或使用场景;

在匹配成功的情况下,提示大语言模型根据流程插件准备匹配插件的入参并调用API,由所述匹配插件完成学习任务;在未匹配成功的情况下,提示大语言模型根据自建文档库或网络资源完成学习任务。

2. 根据权利要求1所述的方法,其特征在于,所述通过大语言模型对所述学习需求进行语义分析,并与各自定义插件的描述信息进行语义匹配,包括:

所述大语言模型通过语义分析,判断所述学习需求是否与各自定义插件的作用或使用场景匹配;

在匹配成功的情况下,所述大语言模型判断所述学习需求是否与各匹配组件的入参和出参匹配;

其中,所述作用或使用场景要求的匹配度,高于所述入参和出参要求的匹配度。

3. 根据权利要求1所述的方法,其特征在于,自定义插件的描述信息还包括当前插件依赖的第一插件;

所述提示大语言模型根据流程插件准备匹配插件的入参并调用API,由所述匹配插件完成学习任务,包括:

根据匹配插件依赖的第一插件,以及匹配插件入参依赖的第二插件,生成以各插件为节点、以插件间的依赖关系为边有向路由图;

提示大语言模型根据所述有向路由图以及图中各边的与或关系,生成以所述匹配插件为终点的至少一条路由;

从所述至少一条路由中确定优选路由,并根据所述优选路由的方向和经过的各插件构建流程插件;

提示大语言模型利用所述流程插件依次执行所述经过的各插件,从而完成学习任务。

4. 根据权利要求3所述的方法,其特征在于,所述根据匹配插件依赖的第一插件,以及匹配插件入参依赖的第二插件,生成以各插件为节点、以插件间的依赖关系为边有向路由图,包括:

构建空白的有向路由图,并将匹配插件作为目标插件;

从所述目标插件的描述信息中,读取所述目标插件的第一插件;将所述目标插件和第一插件作为节点添加至所述有向路由图,并连接从所述第一插件指向所述目标插件的边;

从所述描述信息中读取所述目标插件的入参,在所述入参未知的情况下,从插件中心寻找以所述入参为出参的第二插件;将所述第二插件添加至所述有向路由图,并连接从所述第二插件指向所述目标插件的边;

分别将所述第一插件和第二插件作为新的目标插件,返回读取所述目标插件的第一插

件和入参的操作,直到最新读取的第一插件为空且最新读取的入参已知。

5. 根据权利要求3所述的方法,其特征在于,所述从所述至少一条路由中确定优选路由,包括:

剔除所述至少一条路由中存在环的路由,避免插件无限循环调用。

6. 根据权利要求3所述的方法,其特征在于,所述从所述至少一条路由中确定优选路由,包括:

如果一节点同时存在于多条路由,确定所述节点在各路由中到所述匹配插件的距离,并将距离近的分支作为优选路由。

7. 根据权利要求3所述的方法,其特征在于,匹配插件有多个;

所述提示大语言模型根据所述有向路由图以及图中各边的与或关系,生成以所述匹配插件为终点的至少一条路由,包括:提示大语言模型根据所述有向路由图以及图中各边的与或关系,生成以一匹配插件为终点的至少一条路由;

所述从所述至少一条路由中确定优选路由,并根据所述优选路由的方向和经过的各插件构建流程插件,包括:剔除所述至少一条路由中存在环的路由,避免插件无限循环调用;从剩余路由中选取经过其它匹配插件的路由,作为优选路由。

8. 根据权利要求3所述的方法,其特征在于,所述根据所述优选路由的方向和经过的各插件构建流程插件,包括:

根据脚本注释,自动识别插件模板中的流程节点脚本,其中,所述流程节点脚本中嵌入有当前节点槽位、前导节点槽位和后续节点槽位;

根据所述优选路由经过的插件数量,将所流程节点脚本扩展为多份;

将各前导节点槽位与其上一流程节点脚本中的当前节点槽位自动关联,将各后续节点槽位与其下一流程节点脚本中的当前节点槽位自动关联;

按照路由方向将所述优选路由经过的各插件API,依次补入各流程节点脚本的当前流程节点槽位,实现各槽位的同时更新。

9. 根据权利要求1所述的方法,其特征在于,

所述语言翻译插件的作用或使用场景为“将学习内容翻译成多种语言”;

所述学习推荐插件的作用或使用场景为“根据学员的学习历史和兴趣,推荐合适的课程、资料或学习路径”;

所述在线测评插件的作用或使用场景为“创建在线测验、问卷调查和评估,用于检测学员的理解程度和知识水平”。

10. 一种基于自定义插件和大语言模型的学习系统,其特征在于,包括:

插件中心,用于存储各自定义插件及其描述信息,其中,所述描述信息包括:插件的入参、出参和API,以及插件的作用或使用场景;

控制模块,用于响应于特定应用的触发操作,将学习需求发送至大语言模型,并提示大语言模型分析是否存在能够满足学习需求的自定义插件;其中,所述特定应用包括智能问答、智能课程和智能组卷中的至少之一;各自定义插件包括语言翻译插件、学习推荐插件、在线测评插件和流程插件中的至少之一,所述流程插件用于协调其它插件的执行顺序;

大语言模型,用于对所述学习需求进行语义分析,并与各自定义插件的描述信息进行语义匹配,其中,自定义插件的描述信息包括:插件的入参、出参和API,以及插件的作用或

使用场景；

在匹配成功的情况下,所述控制模块还用于提示大语言模型根据流程插件准备匹配插件的入参并调用API,由所述匹配插件完成学习任务;在未匹配成功的情况下,所述控制模块还用于提示大语言模型根据自建文档库或网络资源完成学习任务。

基于自定义插件和大语言模型的智能学习交互方法

技术领域

[0001] 本发明实施例涉及智能学习领域,尤其涉及一种基于自定义插件和大语言模型的智能学习交互方法。

背景技术

[0002] LLM(Large Language Model,大语言模型)是一种具备交互能力的大型生成式语言模型,模型可以接受用户输入的自然语言,理解其意思并作出相应的反馈。插件是指一种可以添加到其他软件或系统中,以增强其功能或扩展其特性的软件组件,可以帮助软件开发人员更快地创建功能强大的应用程序,同时也可以为用户提供更多的选择,更具灵活性。

[0003] 现有技术中,在基于LLM的智能问答等应用中,为了实现知识快速扩展,可以提示LLM利用已有的知识插件快速获取答案,无需针对每一领域重复构建知识库。例如,专利CN116483980A公开了一种人机交互方法装置及系统,专利CN116795968A公开了一种基于Chat LLM技术的知识扩展及QA系统,均可以利用插件和LLM实现智能问答。

[0004] 但有些插件的执行依赖于其它插件,尤其在为用户提供智能化学习服务的软件或系统中,由于学习需求的多样性和学习过程的复杂性,涉及到的插件数量往往较多。如何组织各插件有序执行来实现特定功能,是亟待解决的问题。

发明内容

[0005] 本发明实施例提供一种基于自定义插件和大语言模型的智能学习交互方法,以解决上述技术问题。

[0006] 第一方面,本发明实施例提供了一种基于自定义插件和大语言模型的学习交互方法,包括:

[0007] 响应于特定应用的触发操作,将学习需求发送至大语言模型,并提示大语言模型分析是否存在能够满足学习需求的自定义插件;其中,所述特定应用包括智能问答、智能课程和智能组卷中的至少之一;各自定义插件包括语言翻译插件、学习推荐插件、在线测评插件和流程插件中的至少之一,所述流程插件用于协调其它插件的执行顺序;

[0008] 通过大语言模型对所述学习需求进行语义分析,并与各自定义插件的描述信息进行语义匹配,其中,自定义插件的描述信息包括:插件的入参、出参和API,以及插件的作用或使用场景;

[0009] 在匹配成功的情况下,提示大语言模型根据流程插件准备匹配插件的入参并调用API,由所述匹配插件完成学习任务;在未匹配成功的情况下,提示大语言模型根据自建文档库或网络资源完成学习任务。

[0010] 第二方面,本发明实施例提供了一种基于自定义插件和大语言模型的学习系统,包括:

[0011] 插件中心,用于存储各自定义插件及其描述信息,其中,所述描述信息包括:插件的入参、出参和API,以及插件的作用或使用场景;

[0012] 控制模块,用于响应于特定应用的触发操作,将学习需求发送至大语言模型,并提示大语言模型分析是否存在能够满足学习需求的自定义插件;其中,所述特定应用包括智能问答、智能课程和智能组卷中的至少之一;各自定义插件包括语言翻译插件、学习推荐插件、在线测评插件和流程插件中的至少之一,所述流程插件用于协调其它插件的执行顺序;

[0013] 大语言模型,用于对所述学习需求进行语义分析,并与各自定义插件的描述信息进行语义匹配,其中,自定义插件的描述信息包括:插件的入参、出参和API,以及插件的作用或使用场景;

[0014] 在匹配成功的情况下,所述控制模块还用于提示大语言模型根据流程插件准备匹配插件的入参并调用API,由所述匹配插件完成学习任务;在未匹配成功的情况下,所述控制模块还用于提示大语言模型根据自建文档库或网络资源完成学习任务。

[0015] 本实施例提供了一种支持自定义插件集成的智能学习交互方法,允许用户根据一定的规则来添加插件,通过自定义插件拓展模型的功能,丰富LLM的使用场景和范围。插件的配置文件存储了关于插件的描述信息,包括插件相关的API出入参、作用使用场景等关键信息。LLM通过对配置文件进行语义分析,就可以获取插件可以完成什么、什么时候该调度,以及如何完成要完成的事情,从而完成插件的调度、调用,实现LLM和自定义插件的结合。当学习需求依赖多个插件共同完成时,本实施例通过流程插件定义自动化流程,协调各插件有序执行,得到满足需求的智能学习结果。

附图说明

[0016] 为了更清楚地说明本发明具体实施方式或现有技术中的技术方案,下面将对具体实施方式或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图是本发明的一些实施方式,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0017] 图1是本发明实施例提供的一种基于自定义插件和大语言模型的学习系统的结构示意图;

[0018] 图2是本发明实施例提供的一种基于自定义插件和大语言模型的学习交互方法的流程图;

[0019] 图3是本发明实施例提供的一种有向路由图的示意图;

[0020] 图4是本发明实施例提供的构建流程插件的示意图,起哄,图4(a)为流程插件的脚本模板,图4(b)为将流程节点脚本扩展为多份后的中间脚本。

具体实施方式

[0021] 为使本发明的目的、技术方案和优点更加清楚,下面将对本发明的技术方案进行清楚、完整的描述。显然,所描述的实施例仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动的前提下所得到的所有其它实施例,都属于本发明所保护的范围。

[0022] 在本发明的描述中,需要说明的是,术语“中心”、“上”、“下”、“左”、“右”、“竖直”、“水平”、“内”、“外”等指示的方位或位置关系为基于附图所示的方位或位置关系,仅是为了便于描述本发明和简化描述,而不是指示或暗示所指的装置或元件必须具有特定的方位、

以特定的方位构造和操作,因此不能理解为对本发明的限制。此外,术语“第一”、“第二”、“第三”仅用于描述目的,而不能理解为指示或暗示相对重要性。

[0023] 在本发明的描述中,还需要说明的是,除非另有明确的规定和限定,术语“安装”、“相连”、“连接”应做广义理解,例如,可以是固定连接,也可以是可拆卸连接,或一体地连接;可以是机械连接,也可以是电连接;可以是直接相连,也可以通过中间媒介间接相连,可以是两个元件内部的连通。对于本领域的普通技术人员而言,可以具体情况理解上述术语在本发明中的具体含义。

[0024] 本发明实施例提供一种基于自定义插件和大语言模型的智能学习交互方法。为了说明该方法,优先对大语言模型的技术原理进行说明。大语言模型为由输入文字得到输出文字的深度学习模型,通过大量文本数据训练得到,可以生成自然语言文本或理解语言文本的含义。传统的语言模型往往面向某一类自然语言任务,如文本分类、翻译、问答等,LLM将模型规模放大,展现了更强的自然语言处理能力(如可以通过上下文进行学习)。示例性的,本实施例的LLM采用T0、ChatGLM、Alpaca、GPT (Generative Pre-Trained Transformer)、ChatGPT等。其中,ChatGPT在LLM的基础上改为对话形式,通过在交互过程中给出提示指令的方式,使用一个LLM来完成多类任务。更具体的,语言模型可以当作一个黑盒,其接受token串作为输入(这里token可以是中文的汉字、或是英文单词等),并输出一个概率,表示此token串是正常人类句子(或片段)的概率。数学形式化如下:给定token序列 (u_1, u_2, \dots, u_n) ,语言模型输出概率 $p(u_1, u_2, \dots, u_n)$,表示了这些token按照顺序组成一个句子(或片段)的概率。如下公式表达了上述的语言模型,将此概率展开成条件概率的形式: $p(u_1, u_2, \dots, u_n) = p(u_1) \prod p(u_i | u_1, u_2, \dots, u_{i-1})$ 。上述语言模型可完成文本生成的任务:给定前面若干个生成好的单词,计算可使序列概率最大的下一个单词,并输出此单词为预测结果;然后模型将预测的单词加入到给定序列,并重复上述过程,继续预测下一个,直到预测下一个单词为结束符号或达到要求长度。

[0025] 在一具体实施方式中,假设输入大语言模型的文字为text,经过下列步骤处理后,即可得到最终的结果output。

[0026] 步骤一、输入预处理。将输入的文字text进行预处理,包括分词、去除停用词、词性标注等操作,得到一个经过预处理的文本序列。假设输入的文字为text,经过预处理后得到的文本序列为tokens,其中每个token表示一个词或一个符号,如表1所示:

[0027] 表1

[0028] 中文分词	text -> tokens	将text分割成一系列的token,用空格符或标点符号作为分隔符,得到tokens序列。例如,对于输入的text为“我是一名学生”,中文分词后得到的tokens为[“我”, “是”, “一”, “名”, “学生”]。
停用词去除	tokens -> tokens_without_stopwords	根据停用词表,去除tokens序列中的停用词,得到去除停用词后的tokens序列。例如,对于tokens为[“我”, “是”, “一”, “名”, “学生”],如果停用词表中包含单词“是”和“一”,则去除停用词后得到的tokens_without_stopwords为[“我”, “名”, “学生”]。

词性标注	tokens_ without_ stopwords - > pos_tags	对tokens_without_stopwords序列中的每个token进行词性标注,标注结果是一个由词性标签组成的序列。例如,对于tokens_without_stopwords为["我", "名", "学生"],进行词性标注后得到的pos_tags为["pronoun", "noun", "noun"],其中"pronoun"表示代词,"noun"表示名词。
------	--	--

[0029] 步骤二、输入编码。将输入的经过预处理的文本序列编码成一个数值向量,以便输入到神经网络中进行计算。可以使用词嵌入技术将每个词映射为一个实数向量,然后将整个文本序列表示为一个矩阵,如表2所示。

[0030] 表2

[0031] 输入编码	tokens -> embedding_ matrix	将tokens序列编码成一个数值向量,以便输入到神经网络中进行计算。通常使用词嵌入(word embedding)技术将每个token映射为一个实数向量,然后将整个tokens序列表示为一个矩阵。例如,对于tokens为["我", "名", "学生"],使用词嵌入技术将每个token映射为一个实数向量,得到embedding_matrix,其中每行表示一个token的词嵌入向量。需要注意的是,中文的词嵌入技术与英文有所不同,通常使用的是基于汉字的词向量或者基于拼音的音节向量。
-------------	-----------------------------------	--

[0032] 步骤三、模型计算将向量化的文本序列输入到大语言模型神经网络中进行计算,得到一个输出向量,如表3所示。示例性的,大语言模型可以使用循环神经网络或变种的Transformer模型进行计算,其中模型参数已经在训练阶段通过大量的文本数据训练得到。

[0033] 表3

[0034] 模型计算	embedding_ matrix -> output	将整个tokens序列表示为一个矩阵后,可以将其输入到循环神经网络(RNN)或变种的Transformer模型中,该网络会根据此输入不断地生成输出token序列,直到生成结束或达到长度要求。
-------------	-----------------------------------	---

[0035] 步骤四、输出解码。将模型计算得到的输出向量进行解码,得到最终的输出文字output_encoding。示例性的,解码可以使用一个输出层将输出向量映射为一个词汇表中的词或字符,然后将这些词或字符组合成一段文字输出,如表4所示。

[0036] 表4

[0037] 输出解码	output -> output_ encoding	将模型计算得到的输出向量output解码成一个数值向量,以便进行输出后处理。通常使用softmax函数将输出向量映射为一个概率分布,然后根据该概率分布选择最可能的词或字符作为输出。例如,对于输出向量output为[0.1, 0.2, 0.7],使用softmax函数后得到的概率分布为[0.22, 0.27, 0.51],则根据该概率分布选择概率最大的第三个元素"0.7"对应的词或字符作为输出。(实际上输出向量的维度会更大)
-------------	-------------------------------	---

[0038] 步骤五、输出后处理。对输出的文字output进行后处理,包括去除多余空格、标点符号等操作,以及根据需要进行进一步的文本处理和分析,如表5所示。需要说明的是,以上内容仅用于说明LLM的技术原理,在实际应用中,可以根据需要开发具体的LLM结构,也通过API调用成熟的LLM算法(例如百度、阿里、讯飞等提供的LLM),本实施例不作具体限制。

[0039] 基于以上LLM,本申请提出一种基于自定义插件和大语言模型的学习系统,该系统的结构如图1所示,包括应用端、控制模块、大语言模型(或接口)、插件中心、本地文档库和

网络资源接口。其中应用端用于向用户提供特定的学习类应用,包括智能问答、智能课程和智能组卷等,各应用由前端界面和后台服务共同实现,集成于所述学习系统内。插件中心内存储有各种自定义插件,包括功能类插件和流程插件;其中,功能类插件用于提供特定的学习功能,例如语言翻译插件、学习推荐插件、在线测评插件,各插件可以理解为多样化学习的最小功能单元;流程插件用于协调其它组插件的执行顺序,通过构建自动化流程将不同的插件和组件整合在一起,以实现更复杂的学习任务。本地文档库和网络资源接口分别为两种与插件中心并列的知识来源,大语言模型在理解学习需求后,可以根据情况选择利用插件中心、本地文档库或网络资源接口来完成学习任务。控制模块用于提示大语言模型执行具体的操作,还用于构建针对具体学习需求的个性化流程插件,不断完善插件中心。

[0040] 基于以上系统架构,图2是本发明实施例提供的一种基于自定义插件和大语言模型的学习交互方法的流程图。该方法适用于某一插件的执行依赖于其它插件的情况,由上述系统中的控制模块执行,该控制模块可以为独立的电子设备、也可以集成于电子设备内的软件模块,本实施例不作具体限制。如图2所示,该方法具体包括如下步骤:

[0041] S110、响应于特定应用的触发操作,将学习需求发送至大语言模型,并提示大语言模型分析是否存在能够满足学习需求的自定义插件。

[0042] 如上所述,本实例可以将各类学习应用展示于前端界面供用户选择,用户可以通过触发按钮或提问的方式,向控制模块提出个性化的学习需求,例如在“智能课程”的应用下提出“请针对我在外文科技论文方面的短板,帮我排一个课表,优先选择比较近的线下课程”。控制模块接收到该需求后,提示LLM优先选用插件中心内的自定义插件对所述学习需求进行响应。示例性,控制模块可以向LLM发出如下提示:

[0043] 用户的学习需求是“请针对我在外文科技论文方面的短板,帮我排一个课表,优先选择比较近的线下课程”,请你对插件中心内的插件进行分析,并返回能够满足学习需求的插件。

[0044] S120、通过大语言模型对所述学习需求进行语义分析,并与各自定义插件的描述信息进行语义匹配。

[0045] 插件中心内的插件包括自行开发的插件,还包括外部开源插件。各插件的配置文件包括对当前插件的描述信息,LLM可以对上述提示进行语义理解,并将其与插件的描述信息进行匹配。可选的,当前插件的描述信息包括当前插件的入参(输入参数或输入信息)、出参(输出参数或输出信息)和API(Application Programming Interface,应用程序编程接口),当前插件的作用或使用场景,以及当前插件依赖的其它插件。示例性的,以下为天气插件的配置文件:

```
[0046] { "name": "WeatherPlugin", #当前插件的名称:天气插件  "version": "1.0.0", #当前插件的版本号  "author": "Your Name", #当前插件的作者信息  "description": "A plugin to retrieve weather information for a specific location and date.", #当前插件的简要描述,包括了插件的作用或使用场景:天气插件用于获取指定地点和日期的天气信息
```

```
"dependencies": {  "geolocation-plugin": "1.0.0",  "date-plugin": "2.0.0" }, #当前插件依赖的其他插件:地理位置插件和日期插件  "entry_point": "getWeather", #当前插件的主要功能入口点(在这里是getWeather) }
```

```
inputs": [{"name": "location", "description": "The location for which
you want weather information.", "type": "string"}, {"name": "date",
"description": "The date for which you want weather information.",
"type": "date"}], #当前插件的入参,包括入参的名称、描述和类型(在这里有location和
date两个入参) "output": {"name": "weatherData", "description": "The
weather data for the specified location and date.", "type": "json"}, #当前
插件的出参,包括出参的名称、描述和类型(在这里是weatherData) "permissions": {"
required": ["geolocation", "date"]}, #当前插件需要的权限或依赖项(在这里需要
geolocation和date权限) "license": "MIT" #当前插件的许可证信息(在这里是MIT许
可证)}
```

[0048] 其中,entry_point为插件的入口点,指插件中的具体的方法或者函数(天气插件中为getweather),在配置文件中会配置这个函数的定义,其中包括API信息。以下是python代码的实现示例:

```
import requests
[0049] def getWeather(location, date):
[0050]     # 构建API请求
[0051]     api_url = "https:#exampleweatherapi.com/api/weather"
[0052]     params = {"location": location,
[0053]                "date": date.strftime("%Y-%m-%d")} # 将日期格式化为字符串,例
如:2023-11-06}
[0054]     try:
[0055]         # 发送API请求
[0056]         response = requests.get(api_url, params=params)
[0057]         # 检查响应状态码
[0058]         if response.status_code == 200:
[0059]             # 解析JSON响应
[0060]             weather_data = response.json()
[0061]             return weather_data
[0062]         else:
[0063]             # 处理错误情况
[0064]             return {"error": "Unable to retrieve weather data."}
[0065]     except Exception as e:
[0066]         return {"error": str(e)}
```

[0067] 基于上述示例的配置文件,控制模块可以通过一定提示使LLM对该文件进行语义分析,从而得知该插件能够完成什么、什么时候调度,以及如何完成要完成的事情。在一具体实施方式中,首先,控制模块提示所述大语言模型进行语义分析,判断所述学习需求是否与各自定义插件的作用或使用场景匹配。插件的作用或使用场景有助于快速判断当前插件是否能够满足应用需求,可选的,语言翻译插件的作用/使用场景为“将学习内容翻译成多种语言”,学习推荐插件的作用/使用场景为“根据学员的学习历史和兴趣,推荐合适的课程、资料或学习路径”,在线测评插件的作用/使用场景为“创建在线测验、问卷调查和评估,

用于检测学员的理解程度和知识水平”。

[0068] 在学习需求与插件的作用或使用场景匹配成功的情况下,控制模块继续提示所述大语言模型判断所述学习需求是否与各匹配组件的入参和出参匹配。入参和出参分别代表了插件执行的前提和结果,也需要达到一定的匹配度。考虑到LLM本身具有一定的信息加工能力,即便学习需求中实际已有的数据与插件入参与有一定差异,或学习需求中实际期望的结果与插件出参有一定差异,LLM也能够将其进行一定程度的调整,则可以将学习需求与插件作用或使用场景匹配度要求,设置为高于学习需求与插件出入参的匹配度要求。示例性的,在将文本向量化表示的情况下,可以通过向量距离来表征匹配度;这时可以提示LLM在确定学习需求(向量表示为R)与插件作用或使用场景(向量表示为F)是否匹配时,如果R与F的向量距离大于第一阈值V1,则认为二者匹配;而在确定学习需求与插件出入参(向量表示为P)是否匹配时,如果R与P的向量距离大于第二阈值V2,则认为二者匹配;同时, $V1 > V2$ 。

[0069] S130、在匹配成功的情况下,提示大语言模型根据流程插件准备匹配插件的入参并调用API,由所述匹配插件完成学习任务;在未匹配成功的情况下,提示大语言模型根据自建文档库或网络资源完成学习任务。

[0070] 根据匹配结果的不同,控制模块为LLM提供两种知识处理方式:由于自定义插件均是根据学习系统需求特别构建或筛选的,更能满足用户的学习需求,因此优先利用自定义插件完成学习任务;如果无可利用的插件,再根据本地自检的文档库或网络资源完成学习任务,以保证对学习需求及时响应。进一步的,使用插件时LLM执行计划根据插件描述、插件出入参等信息构建信息传递给模型,模型根据以上信息对用户的输入进行识别槽位、填补槽位等操作,最终将识别的参数传递给API完成插件的执行。

[0071] 由于学习需求的多样性和学习过程的复杂性,匹配插件的执行可能会涉及到多个其它插件。具体的,在匹配插件的描述信息中明确指出了该插件需要依赖的一部分其它插件(为了便于区分和描述,将该插件称为第一插件);同时,如果匹配插件的入参是未知的,获取入参信息还可能依赖另一部分其它插件(称为第二插件),而第一插件和第二插件的执行又可能依赖又一部分插件(称为第三插件),如此循环往复,涉及到的插件数量可能会很多。其中,第一插件、第二插件和第三插件之间可能存在重复,本实施例不作具体限制。针对这一情况,本实施例提供了一种用于协调多个插件有序执行的流程插件,通过多个插件的有序配合共同完成复杂的学习任务。在一具体实施方式中,上述过程可以包括如下步骤:

[0072] 步骤一、控制模块根据第一插件和第二插件,生成以各插件为节点、以插件间的依赖关系为边有向路由图。所述依赖关系包括插件描述信息中直接提供的依赖关系,也包括获取插件入参时额外引入的依赖关系。通过该有向路由图可以反映插件之间的关联关系,向LLM提供插件网络的全貌信息。

[0073] 可选的,首先,构建空白的有向路由图,并将匹配插件作为目标插件。然后,(S1)从目标插件的描述信息中,读取所述目标插件的依赖的第一插件,将所述目标插件和第一插件作为节点添加至所述有向路由图,并连接从所述第一插件指向所述目标插件的边;边的方向代表了依赖关系中的前导和后续关系。示例性的,如果插件A依赖于插件B,则将插件B称为依赖关系中的前导插件,将插件A称为依赖关系中的后续插件。同时,(S2)从目标插件的描述信息中,读取所述目标插件的入参,在所述入参未知的情况下,从插件中心寻找以所

述入参为出参的第二插件;将所述第二插件作为新的节点添加至所述有向路由图,并连接从所述第二插件指向所述目标插件的边。然后,分别将所述第一插件和第二插件作为新的目标插件,重复S1和S2的操作,如此循环往复,直到某次循环中读取的第一插件为空,终止对S1的循环;直到某次循环中读取的入参是已知数据,终止对S2的循环。其中,每次添加节点之前,还可以检查当前插件是否已存在于有向路由图中,如果已经存在,表明第一插件、第二插件或第三插件有重复,则无需重复添加节点,直接在已有节点之间执行边的连接操作,以表征新的依赖关系。也就是说,同一插件在有向路由图中仅对应一个节点。

[0074] 图3是本发明实施例提供的一种有向路由图的示意图,针对S110中的学习需求“请针对我在外文科技论文方面的短板,帮我排一个课表,优先选择比较近的线下课程”而生成。其中,方块代表匹配插件,圆形代表某插件依赖的第一插件,三角形代表某入参依赖的第二插件。如图3所示,LLM返回的匹配插件包括学习推荐插件1和课表排程插件2。学习推荐插件1依赖的第一插件包括语言翻译插件3和热搜课程插件4;学习推荐插件1的入参包括学习历史和学习兴趣,获取学习历史依赖的第二插件包括课程统计插件5,获取学习兴趣依赖的第二插件包括兴趣捕捉插件6;课程统计插件5依赖的第一插件包括语言翻译插件3,兴趣捕捉插件6依赖的第一插件包括用户画像插件7,用户画像插件7的入参用户地点依赖的第二插件包括定位插件8。课表排程插件2依赖的第一插件包括学习推荐插件1(同时也是另外一个匹配插件)、语言翻译插件3和数学规划插件9;课程排程插件2的入参包括用户地点和线下教学地点,获取用户地点的依赖的第二插件包括定位插件8,等等。图3仅显示了有向路由图中的部分插件,仅用于说明优先路由图包括的主要信息;实际应用中,由于学习需求、已有数据、自定义插件等的不同,在每一次需求响应中生成的有向路由图都不完全相同,本实施例不作限制。进一步的,优先路由图中边与边之间还可能存在着与或关系(图3中未显示),示例性的,当某一插件(如学习推荐插件1)依赖于多个插件(课程统计插件5和兴趣捕捉插件6)同时提供参数或信息时,所述某一插件与所述多个插件之间的多条边是与的关系;当某一插件A依赖B插件或C插件均可以执行时,从B到A的边与从C到A的边为或的关系。

[0075] 步骤二、提示大语言模型根据所述有向路由图以及图中各边的与或关系,生成以所述匹配插件为终点的至少一条路由。示例性的,控制模块可以将上述有向路由图及其边的与或关系通过格式化文件提供至LLM,或转换为描述文字提供至LLM,同时向LLM发出如下提示,利用LLM强大的语义理解及分析能力,快速得到以匹配插件为终点的多条路由:

[0076] 已收到你提供的匹配插件,该插件的执行还涉及到文件A(用于存储有向路由图的文件)中记载的其它插件,这些插件之间的相互关系和执行顺序由文件A中的“依赖插件”和“与或关系”部分提供。请你基于文件A分析出用于执行匹配插件的几条路由,每条路由包括依次执行的多个插件或多组插件(存在与关系的插件构成一组),插件之间传递出入参时,你可以对入参或出参进行一定加工,以满足插件的出入参要求。

[0077] 在另一具体实施方式中,还可以由控制模块根据插件之间的依赖关系逐层溯源,确定出至少一条可达路径。该方式仅是控制模块通过节点遍历实现的,无法利用LLM对出入参的理解和加工能力。因此,两种方式得到的可选路由集合可以相同,也可以不同,或部分相同部分不同。

[0078] 步骤三、从所述至少一条路由中确定优选路由,并根据所述优选路由的方向和经过的各插件构建流程插件。具体来说,本实施例根据实际应用中遇到的几种代表性情况,提

供以下几种可选实施方式来确定优选路由：

[0079] 第一种可选实施方式,适用于存在一节点(称为目标节点)同时存在于多条路由的情况,可以确定各路由中目标节点到所述匹配插件的距离,并将距离近的分支作为优选路由。具体来说,当多条路由均利用了同一插件时,表明该插件在本次知识处理中具有一定的重要性,根据这一信息保留经过该插件的路由;同时,该插件与路由终点的距离越近,对插件信息的利用越直接,在无其它知识信息的情况下,避免引入过多的知识处理环节,兼顾知识处理的简捷性。

[0080] 第二种可选实施方式,适用于匹配插件有多个的情况,所述多个匹配插件可以为多选一的关系,也可以为依赖关系。在多选一的关系下,选用任一匹配插件均可以满足学习需求,在生成至少一条可选路由时可以以任一匹配插件为终点,在确定优选路由时则将经过其它匹配插件的路由作为优选路由。这一方式可以通过一条路由同时执行与学习需求匹配的多个插件,兼顾多个插件与学习需求的匹配角度和知识处理角度,实现对已知强相关知识的充分利用。在所述多个匹配插件存在依赖关系时,如图3中的学习推荐插件1和课表排程插件2,可以在生成可选路由时以依赖关系中的后续插件为路由终点,在确定优选路由时则选取经过其它匹配插件的路由作为优选路由,不仅能够通过一条路由同时执行与学习需求匹配的多个插件,还能明确知识处理的终点,保证知识处理的深度,更符合知识加工逻辑。

[0081] 第三种可选实施方式,适用于除上述两种情况以外的其它情况,默认采用最短路径法,将经过的插件数量最少的路由作为优选路由。该方式在任何知识信息的约束下,不再引入过多的处理环节,侧重于提高执行效率。

[0082] 可选的,在采用上述三种方式确定优选路由之前,控制模块还可以先对所述至少一条可选路由进行预处理,剔除其中存在环的路由,避免插件无限循环调用。剔除完毕后,当仅剩余一条路由时,直接将其作为优选路由;当剩余多条路由时,再采用上述三种方式确定优选路由。

[0083] 优选路由确定后,控制模块可以根据优选路由的方向和经过的各插件,构建适用于本次学习需求的流程插件。当忽略插件间的与或关系时,在一具体实施方式中,流程插件的构建过程可以包括如下步骤:首先,获取流程插件的脚本模板,如图4(a)所示(图中xxx表示省略显示的脚本代码),其中包括流程节点脚本,可以通过以“#”开始的脚本注释语义识别。示例性的,在图4(a)所示的脚本中,将“#流程节点脚本”后的最外层大括号内的脚本段作为流程节点脚本,其中嵌入有当前节点槽位、前导节点槽位和后续节点槽位,均可以通过槽位内特定的参数名称予以识别。然后,复制所述脚本模板作为流程插件的初始脚本,并根据所述优选路由经过的插件数量,将识别出来的流程节点脚本扩展为多份,如图4(b)所示(示例性的,图中扩展为4份,表明优选路由经过了4个脚本)。接着,将各前导节点槽位与其上一流程节点脚本中的当前节点槽位自动关联,将各后续节点槽位与其下一流程节点脚本中的当前节点槽位自动关联,如图4(b)中的虚线所示;示例性的,将流程节点脚本中前导节点槽位和后续节点槽位中的参数自动替换为上一流程节点和下一流程节点当前节点槽位中的参数,实现自动关联。最后,按照路由方向将所述优选路由经过的各插件的API,依次补入各流程节点脚本的当前流程节点槽位,实现各槽位(包括前导节点槽位和后续节点槽位)的同时更新。此外,还可以在各流程节点脚本之间添加LLM加工脚本,用于在流程节点之间

传递出入参遇到问题时(如参数名称、格式或类型不一致等),向LLM反馈问题类型和上一流程节点的执行结果,并接收LLM加工后的数据作为下一流程节点的输入;当然,该脚本也可以包括在流程节点脚本内部,本实施例不作具体限制。上述脚本均添加完毕后,得到流程插件完整的可执行脚本。在考虑插件间的与或关系的情况下,上述方法中的流程节点可能包括多个插件,这时可以根据所述多个插件的与或关系,将各流程节点脚本中前导节点槽位和后续节点槽位中的参数名称进行组合,实现节点之间的自动关联。

[0084] 步骤四、提示大语言模型利用所述流程插件依次执行所述经过的各插件,从而完成学习任务。流程插件构建完毕后,控制模块将其加入插件中心,并提示LLM调用该流程插件,由流程插件自动协调优选路由经过的各插件按照路由方向顺序执行。示例性的,可以向LLM发出如下提示,由LLM自动完成学习任务:

[0085] 请调用编号为特定ID的流程插件,协调插件A、B、C、D(优选路由经过的插件)顺序执行,生成对学习需求“请针对我在外文科技论文方面的短板,帮我排一个课表,优先选择比较近的线下课程”的响应结果。当插件A、B、C、D之间传递出入参遇到问题时(如参数名称、格式或类型不一致等),请根据流程插件反馈的问题类型对传递参数进行调整,以满足插件要求。

[0086] 综上所述,本实施例提供了一种支持自定义插件集成的智能学习交互方法,允许用户根据一定的规则来添加插件,通过自定义插件拓展模型的功能,丰富LLM的使用场景和范围。插件的配置文件存储了关于插件的描述信息,包括插件相关的API出入参、作用使用场景等关键信息。LLM通过对配置文件进行语义分析,就可以获取插件可以完成什么、什么时候该调度,以及如何完成要完成的事情,从而完成插件的调度、调用,实现LLM和自定义插件的结合。当学习需求依赖多个插件共同完成时,本实施例通过流程插件定义自动化流程,协调各插件有序执行,得到满足需求的智能学习结果。特别的,当涉及到的插件数量较大、层级较深时,本实施例通过插件路由图描述插件之间复杂的依赖关系,并借助LLM的语言理解和加工能力生成的多条可选的插件路由,再根据不同情况下需要兼顾知识处理角度和/或简化知识处理环节,确定不同的优选路由,自适应的构建流程插件,提高知识加工的有序性和灵活性,增强智能学习系统对复杂学习需求的响应能力。

[0087] 继续参照图1,本发明实施例还提供一种基于自定义插件和大语言模型的学习系统。在系统应用过程中,插件中心用于存储各自定义插件及其描述信息,其中,所述描述信息包括:插件的入参、出参和API,以及插件的作用或使用场景。控制模块用于响应于特定应用的触发操作,将学习需求发送至大语言模型,并提示大语言模型分析是否存在能够满足学习需求的自定义插件;其中,所述特定应用包括智能问答、智能课程和智能组卷中的至少之一;各自定义插件包括语言翻译插件、学习推荐插件、在线测评插件和流程插件中的至少之一,所述流程插件用于协调其它插件的执行顺序。大语言模型用于对所述学习需求进行语义分析,并与各自定义插件的描述信息进行语义匹配,其中,自定义插件的描述信息包括:插件的入参、出参和API,以及插件的作用或使用场景。在匹配成功的情况下,所述控制模块还用于提示大语言模型根据流程插件准备匹配插件的入参并调用API,由所述匹配插件完成学习任务;在未匹配成功的情况下,所述控制模块还用于提示大语言模型根据自建文档库或网络资源完成学习任务。

[0088] 需要说明的是,本实施例与上述方法实施例基于相同的发明构思,适用于上述任

一方法实施例中的限定,具备上述任一方法实施例的有益效果。

[0089] 最后应说明的是:以上各实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述各实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分或者全部技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案。

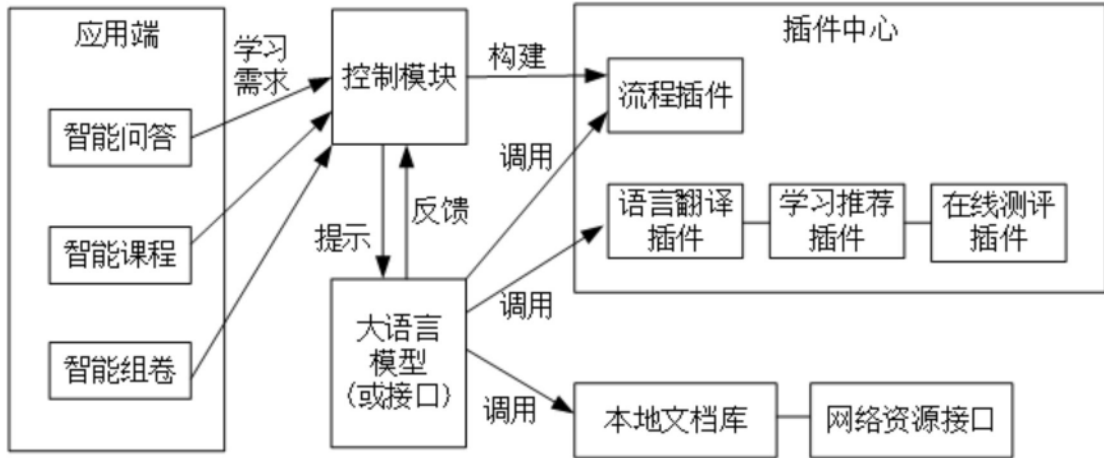


图1

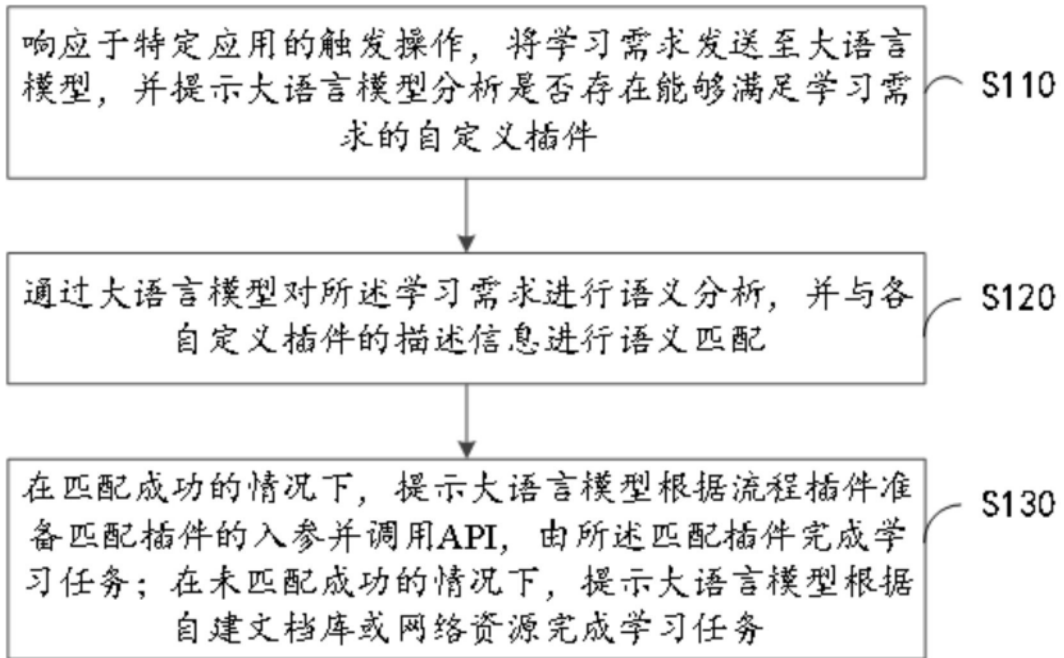


图2

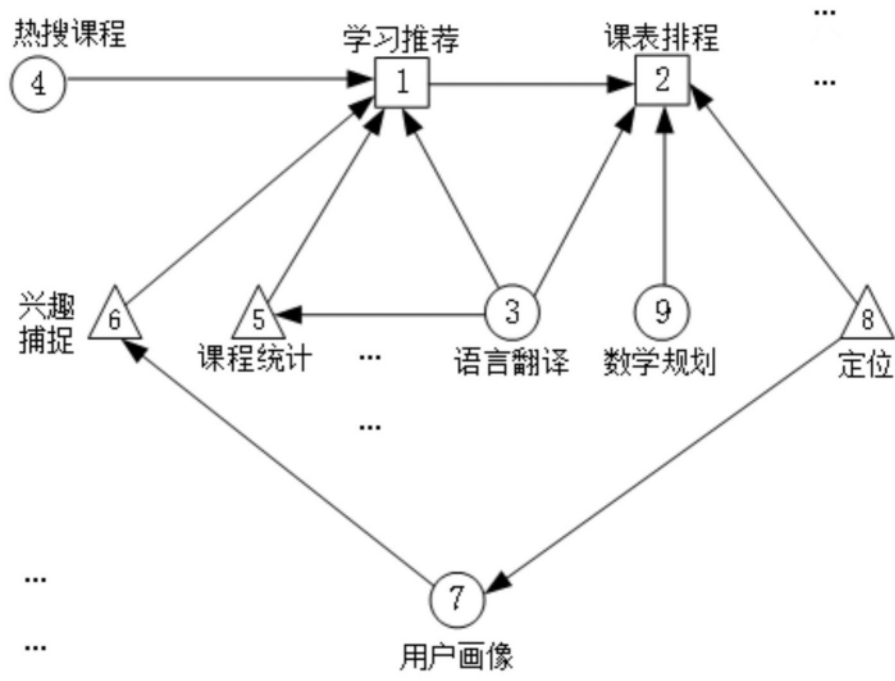


图3

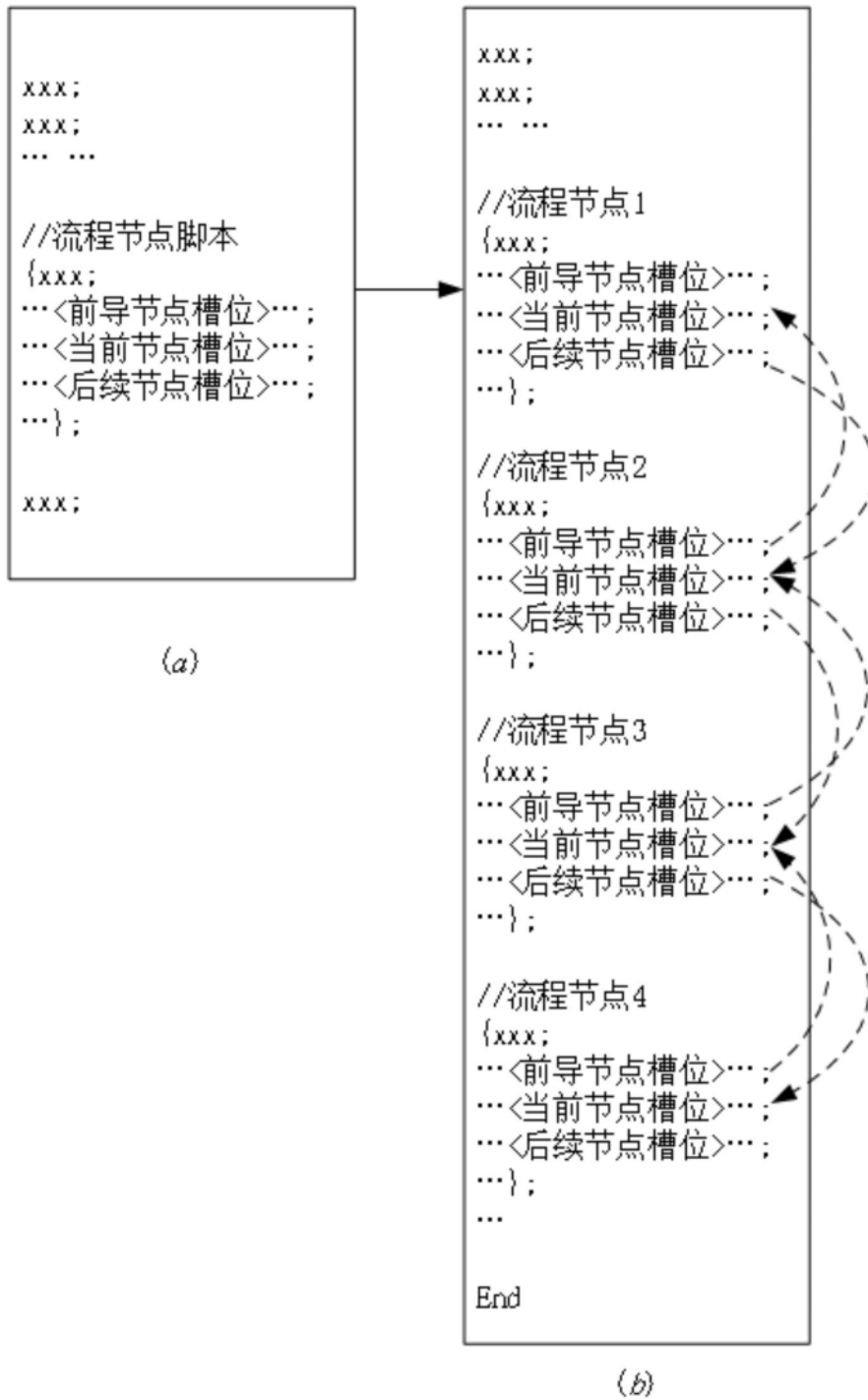


图4